Case Study: EJBs Come to the Rescue

REVIEWED:
• GDPro 5.0

www.java-pro.com
A Fawcette Technical Publication
March 2001, Vol. 5, No. 3

JAVA Pro™

Testing Design Pattern Benefits

Masking Code Complexity

JSP Tag Libraries Hide Unsightly Web Site Code

EPOC OS:
Java Impacts the Wireless World

Field Validation
Made Easy

Simplify
Java Syntax

Free Source Code
Code Online
www.java-pro.com

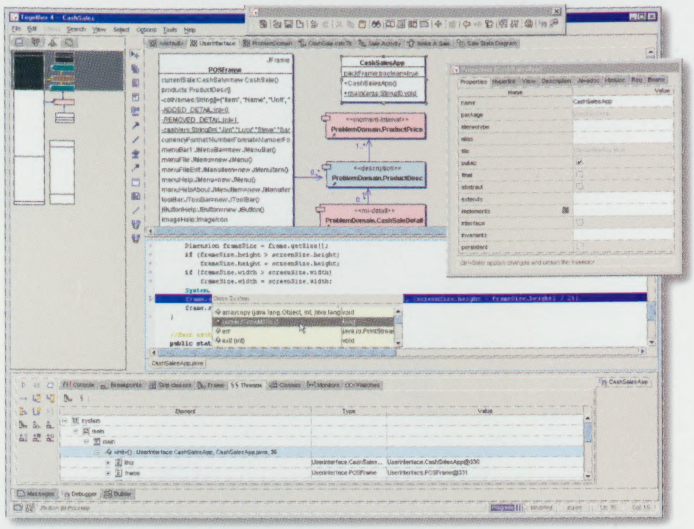
U.S. $5.95 (Canada $7.95)
03>
7 25274 91607 8

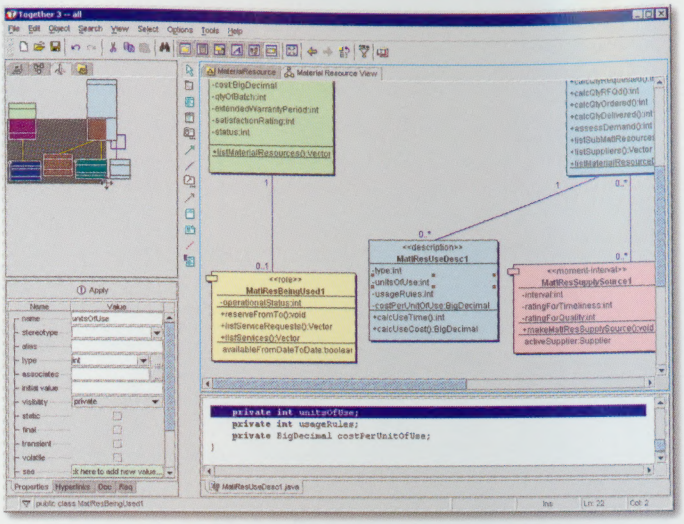
model ➡ edit ➡ gen doc ➡ version ➡ audit/metric ➡ compile ➡ debug ➡ provision ➡ assemble ➡ deploy ➡ run ➡ model ➡ edit ➡ gen doc ➡ version ➡ audit/metric ➡ compile ➡ debug ➡ provision ➡ assemble ➡ deploy ➡ run ➡ model ➡ edit ➡

New! Together Control Center 4.1

Model • Pattern • Edit • Gen Doc • Version • Audit/Metric • Compile • Debug • Provision • Assemble • Deploy • Run

"the backbone of all our software development" —*The Home Depot*



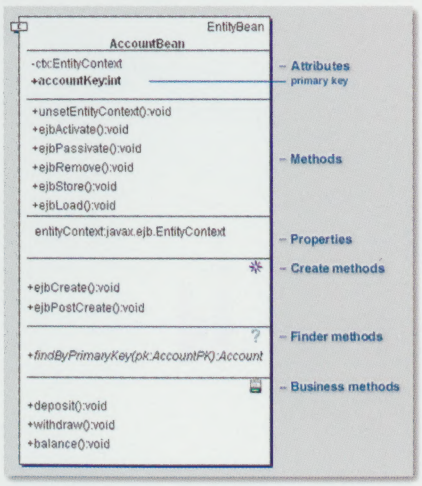


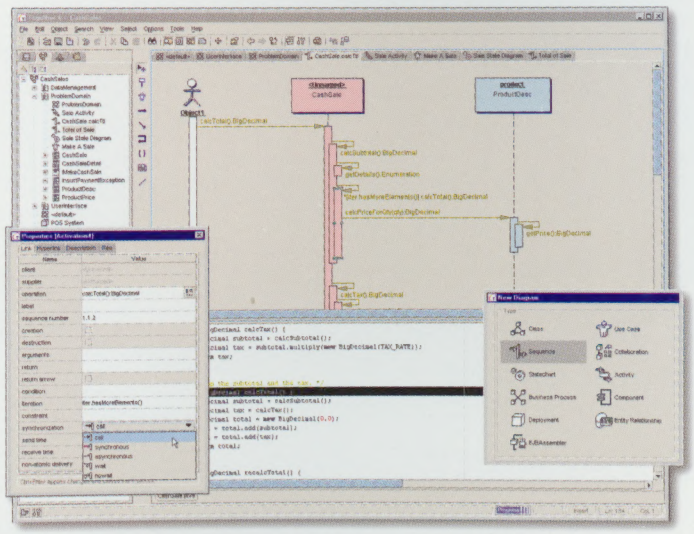
OneSource™ *simultaneous round-trip engineering!* Class diagrams and source code, always in-sync, always up-to-date, giving you complete freedom to work visually or textually (Java, C++ or IDL) as you see fit. Plus robust programming editor with syntax highlighting and auto-completion features; compiler with messages auto-linked to diagram element, source-code line, and property editor; and distributed, multi-threaded debugger.

BigPlay™ *team support is just a mouse-click away!* Complete end-to-end multi-user version control for models, source and documentation across the enterprise (using CVS or whatever SCC-compliant version control system you prefer).

EJB features:

- Generate, configure, reconfigure, customize, and compile
- Visually edit your create, finder and business methods in any EJB
- Together auto-updates the supporting interfaces and primary-key classes (hidden from view, unless you choose otherwise)
- Round trip EJB-JDBC, ERD-JDBC, and JSP-JavaBean support
- Generate, configure, reconfigure and customize GoF patterns
- Add your own company-specific patterns (express simple patterns in text, complex in Java)





UML 1.3 *Nine kinds of diagrams!* Reverse engineer any source code into sequence diagrams, ERDs, too. Plus Raftmaker™ customizable multi-level doc (Javadoc 2 superset + RTF) can be invoked as part of your daily build. Deliver always up-to-date doc to your boss!

Download ***Together Whiteboard*** at **www.togethersoft.com/jp**

"Having seen Together's direct use of Java code to support the models, I am forced to wonder why anyone would like to work in any other way! ...it's a complete solution, not just a modeling tool."
—Jon Collins, Senior Analyst, Bloor Research

"We discovered in our evaluation that other modeling tools only get used during the front end of a project. Yet we found Together to be the one UML modeler that our developers will use from start to finish. It's the only one that delivers simultaneous round-trip engineering (rather than batch mode). Why?"
—Curtis Chambers, The Home Depot

"Thanks for your wonderful product! I wonder how you guys find the time to incorporate so many additional new features in the short span of time between 3.0 and 4.0. I am, as a power user of your product, very happy to see your progress."
— Suresh Balamaran, TimeO

TogetherSoft™

Together Control Center™ Vendor Independence Platform™

TogetherSoft Corporation
920 Main Campus Drive, Suite 410, Raleigh, NC 27606 USA
919-833-5550

©2000 TogetherSoft Corporation, all rights reserved. Not all features are available in all editions.
Together, OneSource, BigPlay, and Raftmaker are trademarks of TogetherSoft Corporation. Java, JavaBean, and EJB are trademarks of Sun Microsystems, Inc.

We tune our J2EE applications with JProbe. Because performance is mission critical.

JProbe™

Speed. Reliability. Rock-solid code. J2EE applications need it all. That's where JProbe comes in. JProbe integrates with all major application servers for fast, accurate analysis of your servlet, JSP and EJB code.

JProbe ServerSide Suite
The ultimate solution for building performance and reliability into your server-side applications.

JProbe Profiler with integrated Memory Debugger
Pinpoint and eliminate performance bottlenecks and inefficient memory usage.

JProbe Threadalyzer
Track down threading hazards including deadlocks, stalls and race conditions.

JProbe Coverage
Don't let untested code put your J2EE apps at risk. Find and quantify untested code to ensure thorough and accurate test runs.

Available in Developer and ServerSide Editions for Windows, Solaris, or Linux. On-site training is also available.

Visit our J2EE integration portal and download a free eval now:

www.sitraka.com/j2ee-javapro

Formerly KL Group 1 888-361-3264 | Europe +31(0)20 510 67 00

sitraka
software
the Java™ advantage

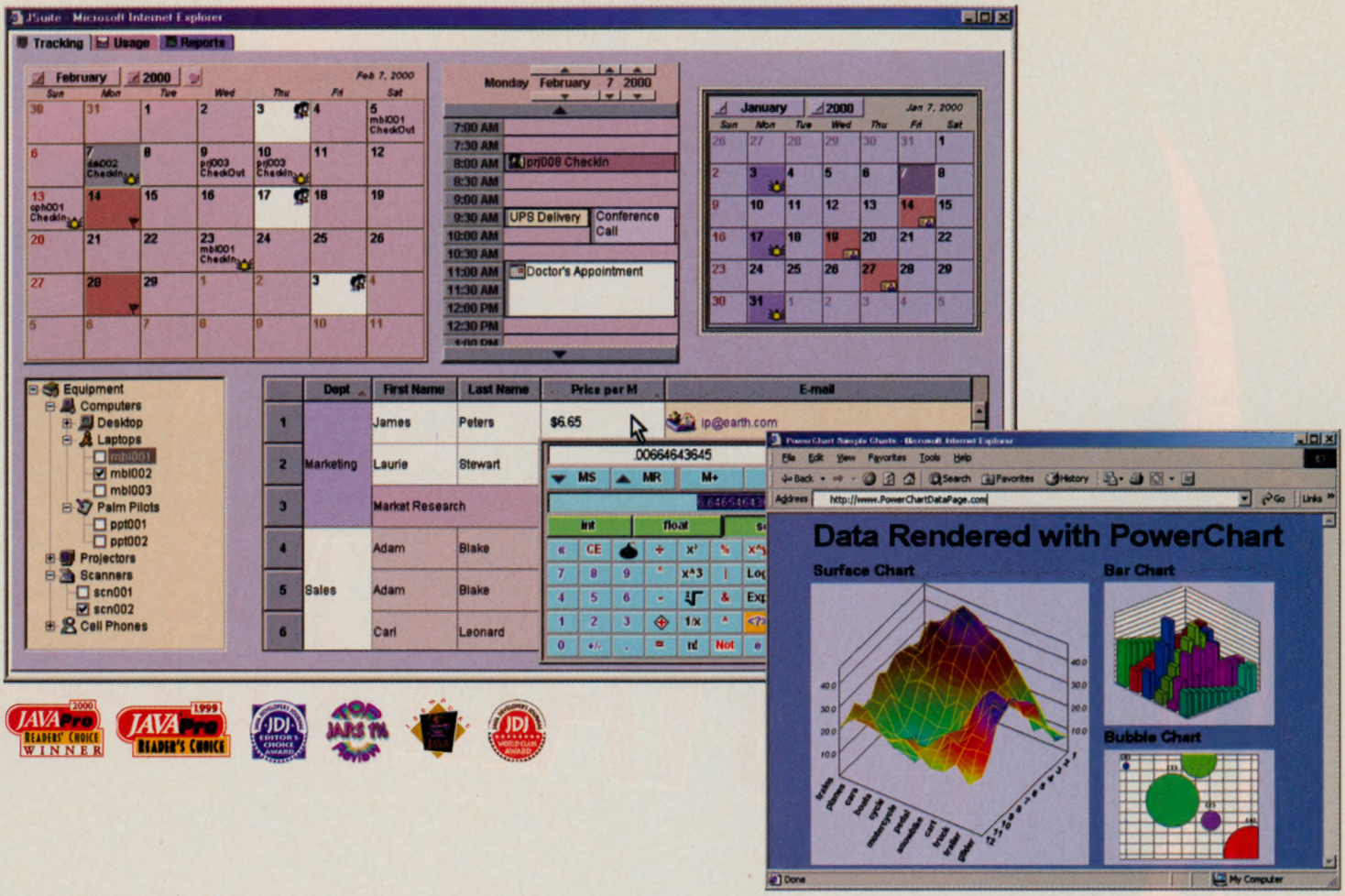
© Copyright 2000 by Sitraka Inc. All rights reserved. Sitraka, the Sitraka logo and JProbe are trademarks of Sitraka Inc. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in US and other countries. All other products are trademarks or registered trademarks of their respective companies.

JFCSuite™ & JSuite™

Finally–Java that looks great!

Add sophistication and visual appeal to your applications with Infragistics' JFCSuite and JSuite. Both the JFCSuite for Java2™ (Swing) and JSuite (AWT and JavaBeans) have everything you need to create powerful, eye-pleasing applications: grid, tree, charting, calendaring and rich graphical data input components, including a **new** drop-down **calculator** and a **color-picker**.

Java • ActiveX • .NET



The **JFCSuite** extends the Swing JTable in ways you may not have thought possible: column locking, vertical and horizontal cell merging, sorting, in-cell validation, printing and much more. The **JSuite** includes a **new DataExplorer** component, for the most comprehensive java application solution available. Use our JSuite components in your web applications–without writing a single line of Java–using JavaScript or VBScript with our **new applet wrappers**.

We've included **PowerChart** so you can render your data in any major chart type– including our **new, scalable bubble** and **scatter charts**–with instantaneous data binding, vivid 2D & 3D rendering and dynamic editing and update capabilities. New features include **server-side rendering*** and hot new **image-mapping** capabilities.

Take advantage of the power and flexibility of java without sacrificing great looks.

* Requires J2EE

Infragistics - formed by the merger of Sheridan and ProtoView

For detailed information, visit
www.infragistics.com
Download Free Trial Version!
Order Online! or call 800-231-8588

Infragistics™
Component powered infrastructure

Copyright 2001 Infragistics, Inc. All rights reserved. Infragistics, the Infragistics logo, JFCSuite, JSuite and PowerChart are trademarks of Infragistics, Inc. All other trademarks or registered trademarks are the respective property of their owners.

InstallAnywhere® 3.5 by Zero G™

Your mission, should you choose to accept it: Powerful, Multi-platform Deployment.

You don't need to be a specially-trained agent to create the world's most powerful deployment solutions. InstallAnywhere from Zero G is the only solution that creates a universal installer that can deploy software from any platform to any platform. And with Zero G's PowerUpdate, it is now extremely simple to add live updating to your applications as well. It's no wonder companies like Sun, Inprise, Adobe, and FedEx count on Zero G to power their deployment. Keep your installers from self-destructing at www.ZeroG.com/ia, or call us at 888-512-7771.

©2000 Zero G Software, Inc. InstallAnywhere, PowerUpdate and Zero G are trademarks or registered trademarks of Zero G Software, Inc.
All other trademarks are property of their respective owners.

zero G

ZERO G

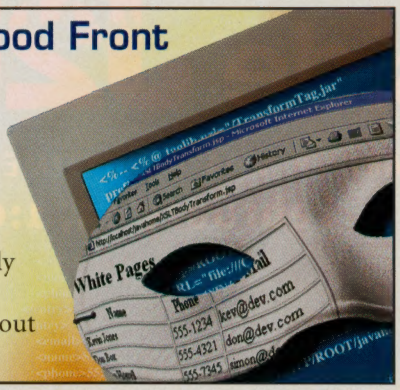
JAVA Pro™

March 2001,
Vol. 5, No. 3

Contents

Features

20 Putting up a Good Front
by Kevin Jones
JSP architects introduced tag libraries to help separate Web page content from Web page logic. The script in tag libraries looks like friendly HTML, but is, in fact, powerful Java code. Find out how it works.



28 Squeezing Java into Smartphones
by Jim Fawcette
Symbian's Executive Vice President David Wood talks with FTP President Jim Fawcette about developing software for a whole new world of smart wireless devices.

34 Networking People
by David Essex and Sue Spielman
Sprint E-Solutions, a network-consulting firm, got bogged down with project management requirements. Find out how a transactional app server with support for EJBs and container-managed persistence solved the problem.

40 A Better Way to Build
by Stuart Halloway
IDEs are effective tools for building Java projects, but things can get confusing when a project has multiple developers. The Jakarta project's open source Ant build tool can eliminate some of these hassles.

45 A Time for Reflection
by Bruce Wallace
Find out how to eliminate burdensome accept() methods from the Visitor pattern using Java 1.2's new reflection capabilities.

Reviews

13 GDPro 5.0
UML modeling tool
76 Book Reviews

Departments

7 Editor's Note
by Steve Gillmor
9 Letters
14 New Products
78 Ad Index
80 Public Static
by Sean Gallagher

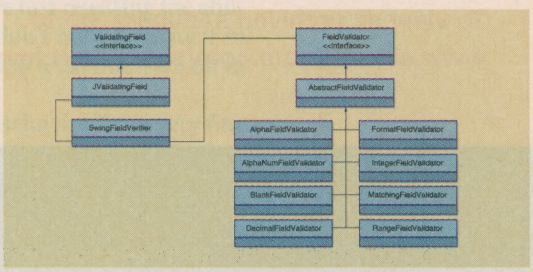
Columns

49 Javatecture
by James Cooper
Researchers have been testing the benefits of design patterns on students and professional programmers alike. Find out what they learned.

53 Visual Components
by Claude Duguay
Find out how a library of text field validators can be flexible and reusable—and can reduce your field validation workload as well.

63 Java To Go
by Rick Grehan
The *Real-Time Specification for Java* defines a set of parameter classes that are, in fact, data classes that can tell us a lot about real-time threads.

71 Pro Shop
by Daniel F. Savarese
Java still has performance problems, and its syntax is more complex than it looks. Find out how to squeeze the best performance out of your programs.



Add field validators to or remove them from your projects using JValidatingField; see page 53.

BPA INTERNATIONAL
PRINTED IN USA

Java Pro *is published monthly—plus a special issue in September—for $34.97 per year, by Fawcette Technical Publications Inc., 209 Hamilton Avenue, Palo Alto, CA, USA, 94301-2500. Tel. 650-833-7100; Fax. 650-321-3818. Send address changes to Java Pro, P.O. Box 54584, Boulder, CO 80322-4584. Application to mail at Periodicals Postage Rates is pending at Palo Alto, CA and additional mailing offices.* © **2001 Fawcette Technical Publications Inc.**, *all rights reserved. Java Pro is an independent publication not affiliated with Sun Microsystems. Sun Microsystems is not responsible in any way for the editorial policy or other contents of the publication. All contents of Java Pro are copyright © 2001 by Fawcette Technical Publications Inc., unless otherwise noted. "VBITS" and "Interactive Developer" are trademarks of Fawcette Technical Publications Inc., a California Corporation, James E. Fawcette, President. Java is a trademark of Sun Microsystems. Rather than put a trademark symbol in every occurrence of other trademarked names, we state that we are using the names only in an editorial fashion with no intention of infringement of the trademark. Although all reasonable attempts are made to ensure accuracy, the publisher does not assume any liability for errors or omissions anywhere in the publication.* **Canadian GST#: 86831 0764**
ISSN: 1096-4495. Ride-Along and Standard Mail Enclosed Presorted Standard US Postage Paid, Permit #17, Waseca, MN 56093.

IMPLEMENTING SOLUTIONS on a J2EE-Certified Platform

GO inside J2EE

MARK THIS DATE AND TIME:
February 15, 1:00 PM EST

INSIDE J2EE Online Seminar Series

NOW WITH VOICE OVER IP

2/15 Introduction to EJB Container-Managed and Bean-Managed Persistence

A short course on the application of CMP and BMP entity beans. The session will review the relative merits of these two types of persistence, and explore and demonstrate their application.

Presented by Michael Demastrie, QA Engineer

Recently Archived Seminars

- Introduction to JavaServer Pages™ (JSP)
- Introduction to Enterprise JavaBeans™
- Integrating SOAP and J2EE™

To view all archived seminars, go to www.silverstream.com/seminararchives.

Attend the INSIDE J2EE online seminar series and learn from industry experts how best to build Java™2 Platform Enterprise Edition (J2EE) Web applications. These 45-minute seminars are for anyone just learning about J2EE, developing J2EE applications, or interested in the future of J2EE. Co-sponsored by Sun Microsystems and SilverStream Software, INSIDE J2EE outlines how and when J2EE technologies are best used, the impact of the platform and how it will best deploy your Web applications.

VISIT THE J2EE ZONE AT SilverStream DevCenter devcenter.silverstream.com

To register, visit www.silverstream.com/onlineseminars or call 1-888-823-9700 (U.S.) or 1-978-840-1765 (outside U.S.).

Register now and enter to win a Palm VII.

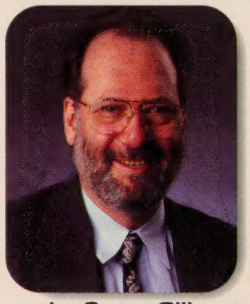
Winner will be chosen from attendees and notified at the conclusion of each seminar.

SilverStream Are you ready to do eBusiness?

Sun microsystems

© 2001 SilverStream Software, Inc. SilverStream is a registered trademark of SilverStream Software, Inc. Sun, Sun Microsystems, the Sun logo, Java, J2EE, Enterprise JavaBeans and JavaServer Pages are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Breakfast with Jeremy



by Steve Gillmor
Editor in Chief
sgillmor@fawcette.com

"Laws are like sausages," said Bismark. "It is better not to see them being made." That's why I ordered bacon when I sat down to breakfast with Jeremy Allaire. The shootout in the application server market has gotten intense as vendors begin to consolidate, but Allaire sees plenty of room at the table for his company's strategy.

"There's probably room for three vendors at the high end of the market. BEA has obviously done extremely well. IBM has done okay. And Microsoft has obviously done very well within the confines of their platform," says Jeremy. And Allaire? "We've focused on a model that makes the technology accessible, on ease of use and abstraction, and, most importantly, on portability."

What exactly are the ingredients of this omelette, I asked? JSPs, tag libraries, and XML, he answered. With Cold Fusion Markup Language, Allaire pioneered a higher-level development environment for dynamic pages and Web applications that Microsoft copied with Active Server Pages and Sun borrowed with JavaServer Pages.

Live Software, co-author with Sun and IBM of the JSP spec, in turn developed "taglets," a tag-extensibility model for Java that became the JSP tag library architecture.

Allaire bought Live, its JRun app server, and its $1,000 price tag ($5,000 with EJB and clustering support). "It was very clear that the marketplace for a high-volume, affordable, easy-to-install-and-use Java server was going to grow really fast."

But aren't these two completely different communities, I asked as I freshened his cup of coffee. "Rich user interfaces, content-rich applications—that's what the Web's all about. And the classic uber programmer, system designer, Java developer—that's not their forté. The tag-language layer is what fuses these two worlds together in an interesting way. It allows you to take complex code and encapsulate it in a declarative way that a designer can just drop into an environment."

Jeremy isn't shy about Allaire's approach to IDEs. "Café, JBuilder, etc., have gobs and gobs of features—JavaBean visual design, Swing design, UI design. We don't give a crap about that. What people want is a clean code-centric IDE environment that doesn't get in their way." Allaire bought Kawa: "a hundred bucks—five hundred bucks for the top end—not these $5,000–$10,000 products that WebGain is selling."

Sounds a lot like the Microsoft model, I ventured as I called for the check. "I don't accept the notion that Dot-Net is just Microsoft copying J2EE...." he said. Microsoft has been way out ahead of the Java community for years in other places. While the Java community was saying they were going to have desktop applets and Java clients—that was just dead wrong. Microsoft was saying, no, you want a browser-based, Dynamic-HTML interface and that's going to be a much richer way to build Web applications."

And then there's XML. "Microsoft, I think, did a really good thing for the industry by mobilizing people around a standard infrastructure—SOAP. We're baking it into our app server—basically, any app you build with—Cold Fusion, JSP, or straight JavaBeans will be able to expose as a Web service, and be able to consume Web services."

People want a clean code-centric IDE environment that doesn't get in their way

For all the Microsoft talk, Jeremy thinks the Java Community Process is working. "Everyone walks a fine line between what they contribute and what they keep on their own—what's standards-based, what's above and beyond that. We've added things that differentiate us. In JRun, we're just releasing the ability to write JSP tags in JSP; you don't have to write it in Java—you just take a JSP page, snap a tag interface on it, and it becomes a tag."

We stand up, and Jeremy leaves the tip: "It's a very fine line. It remains to be seen, in the long term, how much of the software is built on Java. It's proprietary versus based on standards, and it's all contingent on how Sun treats its own software business." ■

Go Online

Use the following DevX Locator+ code at www.java-pro.com to go directly to this related resource.

JP0103EN_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

integrated with System Definition integrated with **Software Development** integrated with Content Management integrated with System Testing integrated with Best Practices integrated with **Software Development** integrated with Change Management integrated with System Definition integrated with **Software Development** integrated with Content Management integrated with System Testing integrated with Best Practices integrated with **Software Development** integrated with Change Management

Developer Controlled-Synchronization of Java code and models. A perfect exchange.

Rational Rose is the #1 Java™ development and Visual Modeling tool. IDC names Rational Rose® the market leader for the fifth year in a row.* Now you can control the synchronization between code and models. Choose the entire model or select model elements. Turn it on all the time or only when you want. Rational Rose is tailored for easy and efficient developer use. It packs the power to help you build better software, faster. Gain full life-cycle support for Java source code including J2SE™, J2EE™ and J2ME™ platforms. Resilient and component-based architecture developed with Rational Rose substantially reduces application rewrites. Put Rational Rose on your team and gain the flexibility, extensibility and compatibility that professional Java developers choose first.

Rational®
the e-development company™

JDJ READERS' CHOICE AWARD · JDJ EDITOR'S CHOICE AWARD · 1999 JAVAPro READER'S CHOICE · Java REPORT 2000 WRITERS' CHOICE

Log-on to www.rational.com/javapro for more information and a Java white paper and CD.

© 2000 Rational Software Corporation. Rational, the Rational logo, Rational the e-development company, Rational Rose, among others, are trademarks or registered trademarks of Rational Software Corporation in the United States and/or other countries. J2SE, J2EE, J2ME, Java and all Java-based trademarks and logos, among others, are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and/or other countries.

*IDC Report 22438: Application Design and Construction Tools Forecast and Analysis, 2000-2004.

JAVA Pro

Jeff Miller, *Vice President, Publishing*
Jeannine Barnard, *Group Publisher, Enterprise Group*

Editorial
Sean Gallagher, *Editorial Director*
Steve Gillmor, *Editor in Chief*
Kay Keppler, *Executive Editor*
Janaya Reitz, *Managing Editor*
Charles Ferrara, *Assistant Editor*

Contributors
James W. Cooper, Claude Duguay
Rick Grehan, Daniel F. Savarese

Art & Production
Michael Hollister, *Vice President, Art & Production*
Bruce Gardner, *Senior Art Director*
Christine Marquez, *Production Director*
Ginger Cross, *Art Director*
Kathleen Sweeney, *Associate Production Manager*
Kim Le, *Senior Designer*
Marine Leroux, *Web Producer*
Julie Ranahan, *Associate Producer*
Ricardo Villarroel, *Associate Producer*
Mona Bedi, *Associate Producer*
Hao Luu, *Web Associate*

Advertising Sales
Shana Deane, *Advertising Director*
Cecile Pendleton, *Account Manager*
David Lyman, *Eastern Regional Sales Manager*
Janice Mahler, *Western Regional Sales Manager*
Beatrice Stonebanks, *Southern Regional Sales Manager*
Jean Anne Mahler, *Eastern Regional Manager, CareerLink*
Kathy Bazan, *Western Regional Manager, CareerLink*
Tami Stoeckle, *Production Coordinator*

Circulation
Meg Clark, *Circulation Manager*
Peter Klehm, *Circulation Director*
Evelyn Adenau, *Associate Circulation Director*
Gerry Guzman, *Circulation Associate*
Joyce Hoelzle, *Circulation Associate*

Marketing
Libby Dickinson, *Marketing Communications Manager*
Margaret Horoszko, *Marketing Designer*
Ray Long, *Marketing Production Coordinator*
Susan Ogren, *Marketing Specialist*
Helga Taylor, *Designer*

Conferences
Tena Carter, *Vice President*
Pam Falk, *Vice President, Conference Sales*
Robert Scoble, *Editor, Web Conferences*
Lee Thé, *Editorial Director*
Maryam Ghaemmaghami, *Associate Editor*
Jennifer Brucker, *Conference Marketing Manager*
Thea Schwartz, *Conference Marketing Director*
Nicole Simonich, *Conferences Marketing Planner*
Shannon Sullivan, *Conference Marketing Planner*
Sarah Lemonds, *Marketing Editorial Assistant*
Cedric Fellows, *Associate Sales Director, Web Conferences*
Heidi Corcoran, *Conference Sales Planner*
Matthew Andress, *Eastern Regional Sales Manager, Windows Group*
Leslie Hasvold, *Conference Operations Planner*
Janet Nickels, *Conference Operations Manager*
Suzanne Nahlik, *Conference Planner*

Customer Service
David Seymour, *Customer Service Manager*
Audra Tate, *Customer Service Manager*
Melanie Meinert, *Customer Service Representative*
Laurie Groenewold, *Customer Service Representative*

Operations
Philip Ziman, *Director of Human Resources and Facilities*
Aaron Wuele, *IS Manager*
Linda Flowers, *Accounting Operations Manager*
Darlyn Philips, *Director of Finance*
Staci Matlack, *A/P Accountant*
P.J. Van Hoy, *A/R Accountant*
Nancy Burgoa, *Staff Accountant*
Iain Neillands, *Staff Accountant*
Leah Spain, *Accounting Assistant*
Neema Buruku, *Financial Analyst*
Pamela Davis, *HR Manager*
Janet Ward, *Senior Human Resource Specialist*
Nancy Wilson, *Office Operations Manager*
Jamie Po'oi, *Facilities Coordinator*
Wendy Gray, *Receptionist*

Fawcette Technical Publications
James E. Fawcette, *President*
Patricia Cameron-Loyd, *Chief Operations Officer*
Jeff Miller, *Vice President, Publishing*
Lynne Matthes, *Group Publisher*
Paul Spyksma, *Publisher, FTP Advertising Network*
Chris Kinsman, *Vice President, Technology*
Wilson, Sonsini, Goodrich & Rosati, *Corporate Counsel*

Letters to the Editor

Letters to Java Pro *are welcome. To be considered for publication, letters must include your name, address, and daytime phone number. Letters may be edited for form, fit, and style. Please send them to Letters to the Editor,* Java Pro, *c/o Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500; fax them to 650-853-0230; or e-mail them to java-pro@fawcette.com.*

Faulty Support Network?

In "Get Your Toes Wet with Open Source" (December 2000), Michiel de Bruijn says, "Despite the popularity of Linux, there is still no official Sun support for this operating system." This is blatantly not true and is hard to believe in an article about how to get going with Java under Linux. Sun has been supplying the Java 2 SDK for Linux since at least December *last* year. (See http://java.sun.com/features/1999/12/linux.html). And version 1.3 has been available since February 2000. I believe Linux is one of only three "tier 1" platforms for Sun. Anyone that has downloaded *any* version of a recent JDK should know this.

To download Solaris, Linux, or Windows JDK 1.3 (listed in that order on Sun's site), see http://java.sun.com/j2se/1.3/.

Craig O'Shannessy

Michiel de Bruijn responds: Craig is 100% right about Sun's Java support for Linux and I sincerely apologize to him and all other Java Pro *readers for this incorrect statement in my article. However, I strongly disagree with Mr. O'Shannessy when he says that "anyone that has downloaded any version of a recent JDK should know this." In the process of researching and writing the article, I visited java.sun.com many times and I definitely got referred to the Blackdown site when I followed the most obvious "Linux" link. I'm not trying to use this as an excuse for including outdated information in the article, but it does at least indicate there may have been some usability issues with the Sun site at that time. Apparently, the JDK download section has since been redesigned, and now indeed prominently features "Linux x86" as a Sun-supported platform, along with Solaris and Microsoft Windows.*

A Dangerous Pattern

I read with interest James Cooper's article about the apparent lack of interest or knowledge among programmers regarding design patterns (Javatecture, December 2000). There is another aspect to this that he should consider: there are programmers, such as myself, whose initial enthusiasm for design patterns has waned considerably over the last few years. The problem is with inexperienced system architects who seem more concerned with design patterns than with solving the problem at hand. Rather than letting patterns inspire our thinking, the patterns substi-

THE PRESSURE TO LAUNCH IS IMMEASURABLE.

WebLoad Resource Manager® can help.

You can test and analyze your site throughout the development process. Allowing teams to communicate faster. Collaborate smarter. And correct problems sooner. So you can deliver your site on time, on spec, and on budget. Because getting it out and getting it right shouldn't be mutually exclusive. Intrigued? Call 888-RadView. Or for a free evaluation, visit www.radview.com.

WebLoad Resource Manager®

RADVIEW™

tute for thinking. Instead of being the start of a design, patterns are the end of it.

Lynne Brown

James Cooper responds: You have a point. Patterns are oversold and underexplained, and thus sometimes adopted as a "magic feather" rather than a considered part of a system design. This month's column ("Testing the Limits," pg. 49) includes research carried out in this area by Professor Walter Tichy.

A Difficult Choice

I congratulate Rick Grehan on "A Scalar Force" (*Guide to Enterprise Java*, Fall 2000). He raised a question that is a major task for every system designer: "If application server A and server B are both J2EE compliant, on what basis will I choose one over the other?" Even after you've expended a lot of effort to come to the conclusion that an application server is the best solution, you still have this most difficult question to answer. I think this question has to be addressed at a global level.

Kishore Pejavara

Rick Grehan responds: Thanks you for your kind note. Good luck in your wrestling matches with application servers; choosing is indeed difficult.

Troublesome IDEs

Reading Michiel de Bruijn's review of the WebGain Studio 4.0 Java IDE (December 2000) prompted me to write with a request. I've started using Java as my primary tool for building applications, switching from VB, C++, ASP, MTS, COM, etc. to an EJB environment. I bought into the "write once run anywhere" goal. In the environment I'm in, I can target Unix, Linux, OS/390, NT, etc. Also, I need to deploy applications to IBM's WebSphere, BEA's WebLogic, and the iPlanet application servers. The problem I'm discovering is deploying applications built in the enterprise version of my IDE of choice to app servers from companies other than the one that makes my IDE. EJBs are particularly troublesome (although I haven't tried JMS and some of the other J2EE components).

I'm writing to ask that in future Java IDE reviews, please make one of the observation points the difficulty or ease of deploying J2EE apps built in the IDE to the major app servers. This is a crucial point for many of us enterprise developers, and we could have used a scoring or at least an observation on this point to help us decide which IDE to purchase. At the enterprise level, these IDEs are expensive and complex. Any help we can get with issues such as this would help us greatly in our decisions and in our productivity. Not having to master a different IDE for each application server environment is a requirement of the market. If the manufacturers of these IDEs and application servers see that being able to deploy a J2EE application to many different environments is important to the market, they may help make our lives a little easier through supplying an add-on or some other such mechanism. It seems ironic that the "write once run anywhere" goal could be replaced by the IDE's "write here and run only where I say" limitations.

Tim Oliver

The editors respond: Thanks for your suggestion. We'll keep this in mind for future reviews of IDEs.

Debugging JavaServer Pages

In "Validating JSP Input with JavaBeans" by Daniel F. Savarese (May 2000), the question was posed: "What are the techniques for debugging JavaServer Pages?" Mr. Savarese indicated there was not a known method to debug JSP pages. Borland's JBuilder 4 has this ability. Here is an excerpt from the Help module: " . . . JBuilder provides a complete development system for JSPs, including a JSP wizard for creating a new JSP, CodeInsight for JSP-specific tags, debugging within the JSP file, and testing and running the JSP on the Tomcat servlet engine from within the JBuilder development environment. . . . " So I am a little baffled by this statement.

Joe Baker

Daniel F. Savarese responds: My exact words were: "JSP is still a young technology and I am as yet unaware of any debugging environments for it. That does not mean no one is working on them. I just don't know about them." This is very different from saying that there is no known method. For that matter, I went on to describe a way to debug in the absence of a debugging tool. Also, remember that the article was published in May 2000, which means it was written in March. JBuilder 4 was released in October of last year, well after the column was written.

Since writing that column, I have become better acquainted with the various commercial Java development environment features. All of the major Java development environments, except for Forte, provide JSP debugging support only in their enterprise editions, which cost several thousand dollars a piece. I can't speak for JBuilder because I haven't spoken with one of their sales representatives, but Visual Café's JSP debugging support references source code in the generated servlet, rather than the JSP page, which makes it of limited usefulness. They're working on improving this for the next version.

Go Online

Use the following DevX Locator+ code at www.java-pro.com to go directly to this related resource.

JP0103LE_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

Ericsson | Echelon | Perkin Elmer | IntelliCorp | KLA-Tencor | BuildNet | Evolve | Lucent | Siemens | US West | IDS Scheer

Looking for a database infrastructure with superior performance and total cost of ownership?

let's talk.

If you're serious about developing eCommerce solutions, don't waste your breath talking to anyone else about Internet infrastructure. Chances are, they'll offer you a piecemeal approach based on the object technology we invented in the first place. So come straight to the source. Where our innovative solutions to meet any Internet challenge are more than just idle chitchat. They're what the rest of the industry is just talking about. Call us at 800-950-8845 or visit our web site at www.poet.com/oss

POET

www.poet.com

Review

GDPro 5.0

Take charge of your development process

by Michiel de Bruijn

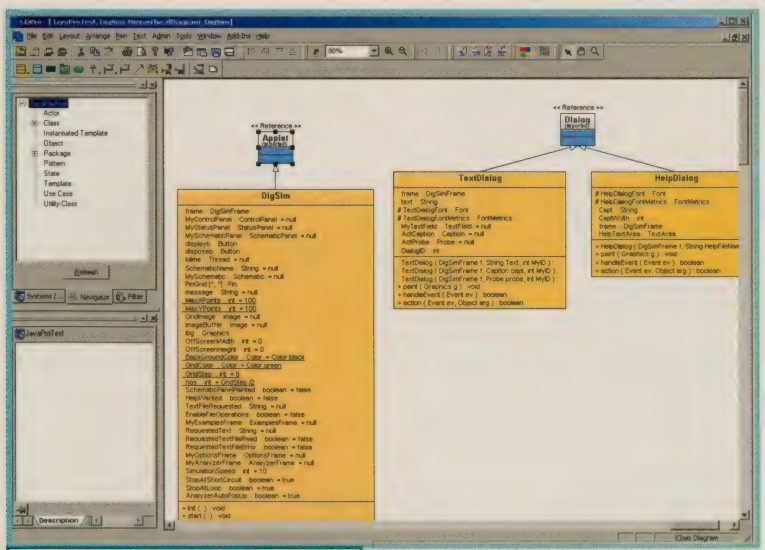
Quick Facts:

Whether a Unified Modeling Language tool is the right solution for your organization's development problems is a bit of a religious issue—but if it is the right solution, GDPro offers solid modeling, round-trip/reverse engineering, and team development features for both Java and C++.

Advanced Software Technologies, Inc.
7851 South Elati Street, Suite 102
Littleton, Colorado 80120
Phone: (303) 730-7981
Web: www.advancedsw.com
Price: $2,995 (Windows version; Unix version $4,795)

A recurrent theme in software engineering literature is how to make complex development processes more controllable, especially in terms of quality of code and schedules, as well as make the processes more transparent for those who manage them. Of all proposed solutions, the combination of object-oriented programming and the Unified Modeling Language (UML) has gotten traction with a wide audience.

Here's how UML is supposed to work: based on user requirements, a formal model is generated, which can be understood by programmers and nonprogrammers alike. This blueprint will then be used for design verification as the basis for implementation, as well as for identifying commonly used classes, which can then be encapsulated in libraries, allowing for re-use of the design and the code.



Reverse Engineering the Easy Way Like most UML tools, GDPro supports the creation of a model based on existing code. Unlike most of the competition, though, GDPro can process entire projects at once without special preprocessing, as well as update the model based on changes in the code without adding special comment tags or markers, keeping it clean and compilable.

To make all this good stuff work in reality, you need a strong set of tools to support a wide range of loosely coupled activities.

Strength Through Flexibility

Advanced Software Technologies' GDPro 5.0 interfaces with both requirement analysis and development tools (although its support for the latter category is the strongest by far) and, unlike many of its competitors, allows for free-style extensions and annotations that don't follow the strict UML rules.

GDPro is also pretty good at keeping the diagrams associated with a model in sync with your (Java or C++) code. It doesn't litter your source with annoying comments or markers and has the welcome option to turn off code generation for certain parts of the model that you never plan to implement anyway.

More welcome features include explicit support for Enterprise JavaBeans (EJB). A wizard-style helper program creates skeleton code for all the required interfaces—even for the XML Deployment Descriptor, which is required by the target application server—based on your model. The JavaDocs standard is fully supported as well.

For those cases where you have existing code that needs to be incorporated in a model, reverse engineering will help you out. Unlike many other tools, GDPro is intelligent enough to import complete projects (including most dependencies that might exist) without manual intervention, although you should count on having enough memory and time available for really large projects. And although it may not always use your screen real estate in the most efficient way possible, the diagrams resulting from reverse engineering have a pleasant layout (see figure).

In an ideal world, project specifications and code don't change after their initial definition. In real life, though, they do so with a vengeance, and this is where GDPro's incremental reverse engineering feature comes in handy. You can refine your model based on new code versions easily and without losing any annotations you may have made earlier.

100% Pure UML

Flexible though it is, GDPro 5.0 still does fairly well in the area of compliance with UML standards. All nine standard UML 1.3 diagrams, as well as all three UML extension mechanisms (stereotypes, tagged values, and constraints) are fully implemented and intuitive to use.

GDPro comes with strong support for team development, as well. Its repository-based approach and integration with most revision control systems should allow for easy integration with any enterprise development environment. Support for Microsoft's VBA means that you can create your own product extensions using Visual Basic or any other language capable of using COM components in automation mode.

IDE support is a bit of a mixed bag, though: Whereas C++ programmers using the Microsoft Developer Studio get some handy tools to extend their environment (not to mention premodeled versions of the MFC and ATL libraries), Java programmers have to do without such goodies.

For now, GDPro 5.0 is one of the most flexible UML tools on the market.

Michiel de Bruijn lives in Rotterdam, The Netherlands, and spends his days as a networking and development specialist for an international media group. He seems to enjoy living on the bleeding edge of technology. Michiel always welcomes questions and comments: reach him at mdb@x42.net.

New Products

E-Commerce

Graphics and Charting

IDEs

Database and Middleware Tools

Application Servers

Authoring and Deployment Tools

Libraries

Developer Tools

JVMs

Services

Testing and Debugging

E-Commerce

jCommerce

jCommerce is a 100% Pure Java hardware-independent Web development application. Its open architecture integrates with multiple platforms (IBM, Sun, Linux, NT, and so on). It also allows users to build a secure, complex, and compatible Web interface using a multitude of back-end applications or legacy systems. Features include real-time access to order, inventory, and financial data; and tax calculation and pricing routines. Contact vendor for pricing.

eOne Group
Phone: 402-970-6750
E-mail: sales@eonegroup.com
Web: www.eonegroup.com

Graphics and Charting

JLOOXLayout 1.3

Designed as an add-on module to JLOOX, JLOOXLayout is a collection of Java classes used to visualize relationships between entities represented by JLOOX and Java Foundation Classes/Swing components. The latest version enables detection and visualization of disconnected subnetworks, automatic visualization of multiple connections between two network nodes, and automatic layout of work- and data-flow diagrams. $2,000.

LOOX Software
Phone: 650-552-9310; 800-684-LOOX
E-mail: info@loox.com
Web: www.loox.com

NetCharts 3.6

This suite of Java applets lets HTML authors, Web developers, and Java programmers create quick, easy charts online. The applets can be viewed through Java-compatible Web browsers and they support the use of JDBC interfaces to retrieve parameter definitions or data values from data sources including Oracle, Sybase, Informix, DB2, Microsoft SQL Server, dBase, Access, and Excel. NetCharts supports any Java 1.0.x-enabled or 1.1.x-enabled platform, including PCs, Unix, and network computers. Starts at $495.

Visual Mining, Inc.
Phone: 301-795-2200; 800-308-0731
E-mail: info@visualmining.com
Web: www.visualmining.com

NetCharts Beans

NetCharts Beans is a tool intended to provide a powerful environment for developing chart-enabled applications. Designed to combine the benefits of software component architecture with the flexibility of Chart Definition Language (CDL), the program enables multiple data sources to be simultaneously charted on a single chart, and charts to be deployed as applets, class libraries, or beans. End users can drill down from anywhere on a chart and the chart automatically scales labels, axis tic marks, fonts, axes, and size. It runs on any Java-supported platform including Unix, Windows NT, Windows, Macintosh, OS/2, Network PCs, and JavaStations. Starts at $995.

Visual Mining, Inc.
Phone: 301-795-2200; 800-308-0731
E-mail: info@visualmining.com
Web: www.visualmining.com

IDEs

Forte for Java Community Edition 1.0

This integrated development environment (IDE) for the Java 2 Platform is based completely on open standards. The modular Java technology development environment features integrated visual design, editing, compilation, and debugging capabilities for creating full-featured Web-enabled applications based on the Java programming language. The program is completely cross-platform compatible, enabling development on the Solaris Operating Environment, Linux, and Windows platforms. Free.

Sun Microsystems Inc.
Phone: 650-960-1300
Web: www.sun.com

JET

Designed to boost Java application performance by allowing Java applications to be deployed as highly optimized native 32-bit Windows executables, this Java 2 deployment environment's core component is an optimizing, Java-to-native code compiler. This compiler reads Java class files created in any Java IDE, compiles them, links with high-performance runtime support libraries, and produces a conventional executable file. Contact for pricing.

Excelsior
Phone: (650) 269 0091
E-mail: info@excelsior-usa.com
Web: www.excelsior-usa.com

WebGain Studio 4 Standard Edition

This Java toolset provides an end-to-end, browser-to-database Java development solution. WebGain Studio helps you quickly build Java component-based solutions that are easily deployed across leading application servers. $4,995.

WebGain, Inc.
Phone: 408-517-3700; 877-932-4246
E-mail: sales@webgain.com
Web: www.webgain.com

Database and Middleware Tools

iPex for Java

This cross-platform XML processing engine incorporates XML functionality into existing applications on Windows, Unix, and Linux. It provides a set of C++ functions and subroutines that allow storage, retrieval, and manipulation of XML data, processing of DTDs and Namespaces, and performance of on-the-fly verification. The program also enables XML to be implemented on Visual Basic applications and Active Server Pages, and is fully compatible with a wide range of languages, including all three Japanese and nine European encoding methods. Starts at $150.

Infoteria USA
Phone: 978-922-4029
E-mail: info@infoteria.com
Web: www.infoteria.com

XMLShark 2.0

XMLShark 2.0 is a 100% Pure Java data integration tool designed to enable rapid exchange of data between enterprise databases and net-based users. It allows developers to generate relational XML

Remember when sharing your design was this simple?

Now team communication can be this easy with GDPro.

Utilizing the industry-standard Unified Modeling Language (UML), GDPro is the quickest way to integrate analysis, design and reengineering into your development process.

GDPro is a complete visual UML modeling solution for accelerating software design and collaborative development. It allows application developers to create and communicate across the enterprise using simple, comprehensive visual blueprints of Java, C++ and IDL. Features include: Markerless round-trip engineering; Inter-model synchronization; Web system reports; VBA customization; Best-of-breed lifecycle tool integrations; and Rose import.

GDPRO

Start sharing your vision today! Visit our website to get a free evaluation copy of GDPro or call (800) 811-2784.

www.gdpro.com

© 2001 Embarcadero Technologies, Inc.
GDPro is a trademark of Embarcadero Technologies, Inc.
All other brand and product names are trademarks or registered trademarks of their respective holders.

New Products

data from legacy sources. The XML data can be securely used, shared, and exchanged on the Web and then parsed and returned to a corporate source. The program requires a Java 2 development environment and Oracle 7, Oracle 8, Oracle 8i, or Microsoft SQL 7 local or remote server. Contact for pricing.
infoShark
Phone: 703-802-8210; 888-328-2742
Web: www.infoshark.com

Allora for Java

HiT Allora for Java is database-focused DOM/SAX implementation. The program is designed to help maintain an XML-consistent interface across all relational databases by allowing both read and write database integration. Developers can use Sun standard JAXP interfaces to build W3C DOM objects, transform them as necessary, update node values, and apply the XML data changes back into the underlying DBMS. It supports JDK 1.2 versions and later. It is available in Java and native Windows versions, and compatible with all major Web application development platforms including BEA WebLogic, IBM WebSphere, iPlanet, SilverStream, and others. Contact vendor for pricing.
HiT Software Inc.
Phone: 408-345-4001
E-mail: info@hitsw.com
Web: www.hit.com

Application Servers

SonicMQ 3.0

This JMS-compliant Java Messaging Server provides developers with a set of features for event-based communications between distributed applications. New to version 3.0 is the Dynamic Routing Architecture, a feature that extends functionality in the areas of scalability, availability, security, reliability, performance, and connectivity and standards. It is certified for the following operating environments: Microsoft Windows NT, Sun Solaris, Linux, HP-UX, and AIX. The program is available in three editions: Developer, Professional Developer, and E-Business Edition. Contact vendor for pricing.
Progress Software Corp.
Phone: 781-280-4000; 800-477-6473
Web: www.progress.com

Pramati Server 2.0

Pramati Server 2.0 is a server-side component development and testing platform with pluggable tools and tool macros in Java. The program's features include: a configurable workspace that allows customizable views on projects and component development, an embedded test server for component lifecycle management, and debugging capabilities for remote VMs and multiple VMs. $1,000 per user.
Pramati Technologies Limited
Phone: 408-965-5513; 877-PRAMATI
E-mail: info@pramati.com
Web: www.pramati.com

Authoring and Deployment Tools

eContent

eContent is a Web-based content management system for creating enterprise information systems. Designed to integrate content management, scalable content delivery, resource management, and personalization, the program features support for managing content stored in databases, XML repositories, and static files. It also includes built-in security, an unlimited table of contents, unlimited n-level categories, and unlimited resources, such as static reports, live reports with drill-downs, executables, OLAP reports, Java programs, online forms, and legacy integration. Servers can be used as distribution points for reports and other resources created on a wide variety of platforms. The program is built using the Expresso Framework and requires Expresso for deployment. $2,000.
Jcorporate
E-mail: info@Jcorporate.com
Web: www.javacorporate.com

Jasper 1.0.2

Jasper 1.0.2 is a COM object that reads and executes code from existing Java classes and converts it to COM objects. This allows Java classes to work transparently as COM objects with ASP, VB,

ONLINE RESOURCES

Locator+ code: JP0103
Type this code in the Locator+ field at the top of the *Java Pro* home page. You can find all the code in the current issue of *Java Pro* at www.java-pro.com. All the listings and associated files discussed in the articles are available free in one ZIP file. To access code in issues from 1998 and 1999, type "JP" followed by the month and year. For example, the Locator+ code for the October 1998 issue is JP1098. To access code from issues published after December 1999, type "JP" followed by the year (two digits) and the month (two digits). For example, the Locator+ code for the January 2000 issue is JP0001.

Premier Club Members

Members of the Premier Club of the Development Exchange (www.devx.com, $24.95 for six months, $44.95 for one year) can get each article's listings and associated files in separate ZIP files. For example, to download the individual code file associated with the October 1998 "Ask the Java Pro" column by Daniel F. Savarese, you would type in JP1098AP as the Locator+ code. For any articles published after December 1999, however, the position of the year (two digits) and the month (two digits) will be switched in the Locator+ code. For example, to download the individual code file associated with the January 2000 "Ask the Java Pro" column, you would type in JP0001AP. Premier Club members can also access archives of all articles and related code ever published in *Java Pro*.

Java Zone

The Java Zone (www.java-zone.com) on the Development Exchange (www.devx.com) offers additional Java resources including product reviews, programming tips, weekly downloads, and discussion forums.

HOW TO REACH US

Editorial Offices:

Fawcette Technical Publications
209 Hamilton Avenue
Palo Alto, CA 94301-2500
Phone: 650-833-7100
Fax: 650-853-0230
Web: www.java-pro.com
E-mail: java-pro@fawcette.com

Article Submission

To submit articles for publication please contact:
Kay Keppler, Executive Editor, *Java Pro*
E-mail: kkeppler@fawcette.com
Download the Author Guidelines at
www.java-pro.com/authguid.asp

Product Reviews

To submit products for review please contact:
Chris Preimesberger, Senior Editor
E-mail: cpreimesberger@devx.com

Product Announcements

Please e-mail press releases to:
pressreleases@fawcette.com

Reprints and Permissions

For all *Java Pro* editorial and advertising reprints contact:
PARS International Corp.
114 East 32nd Street, Suite 503
New York, NY 10016
Phone: 212-221-9595
Fax: 212-221-9195

To quote from an article, please contact:

Tami Stoeckle
209 Hamilton Avenue
Palo Alto, CA 94301-2500
Phone: 650-833-7118
E-mail: tstoeckle@fawcette.com
Specify the issue date and title of the article, the portion you would like to quote, and the purpose.

Photocopy Rights:

Permission to photocopy for internal or personal use may be granted by Fawcette Technical Publications. Please contact Tami Stoeckle for more information.

Customer Service and Subscription Information

For subscription orders, inquiries, or address changes please contact Customer Service:
Java Pro
P.O. Box 54584
Boulder, CO 80322-4584
Phone: 303-945-5282
Fax: 650-321-3818
Foreign and Canadian orders must be payable in U.S. dollars, plus postage. The surface rate to Canada and Mexico is $52.97 per year. For all other countries the air mail rate is $78.97 per year.

Back Issues

To order *Java Pro* back issues, call 650-833-7100 or 800-848-5523 and ask for Customer Service. Back issues cost $10. Additional postage will be charged to deliveries outside the USA.

What did one distributed application say to the other?
We need to talk.
Progress.
SonicMQ™
The world's fastest E-Business Messaging Server
Performance. Not Promises.

© Copyright 2001 Progress Software Corporation. All rights reserved. Progress and all other names of Progress Software Corporation products are registered trademarks or trademarks of Progress Software Corporation. All other trademarks and registered trademarks, whether marked or not, are property of their respective owners

Are your distributed applications on speaking terms? Now there's a solution that guarantees the reliable exchange of information: Progress® SonicMQ™. This 100% Java, fully JMS-compliant, Internet messaging server includes everything you need to enable event-based communication between business-critical applications — whether you use a point-to-point or broadcast model. You can even send XML formatted messages. Need to know more about this standalone, scalable solution? **Call 1-800-989-3773, ext. 325, mail the attached reply card, or visit www.sonicmq.com/jp301.htm for a FREE Developer's CD and information.** Then let's talk.

PROGRESS SOFTWARE

SonicMQ: Get the message!

Sun microsystems

New Products

Delphi, Centura Team Developer, VBA (Word, Excel), and any other COM-compliant tool. It works automatically with Microsoft JVM and Sun JVM and supports overloaded functions, constructors, arrays, static members, and virtual classes. The program is also thread-safe, supporting overloading and the ability to catch or throw real Java exceptions. $120.
Ice Tea Group
Phone: 301-765-2276
E-mail: manager@iceteagroup.com
Web: www.iceteagroup.com

Together Control Center 4.1

Together Control Center 4.1 is designed to accelerate e-business and enterprise application development for teams using Java and Java 2 Enterprise Edition (J2EE) application servers. The program also supports teams using C++, IDL, and XML, delivering wider coverage and support for large development organizations. New features include: an XML structure editor, an expanded EJB Deployment Expert, GoF patterns for C++, plus additional C++-specific patterns, enhanced audit and metrics, improved compile/make/run/debug, improved forward and reverse engineering of database schemas, and a WLE CORBA Expert for working with BEA WebLogic Enterprise. Contact vendor for pricing
TogetherSoft LLC
Phone: 919-833-5550
E-mail: info@togethersoft.com
Web: www.togethersoft.com

Libraries

Expresso Framework

Designed to allow developers to build entire systems, Expresso Framework is a library of Java servlet components for providing services to Web-based client applications. The program features secure, high-performance database access via an object-to-relational mapping, background job queuing and handling with dynamic invocation of custom server-side objects, and sophisticated event notifications and logging. Contact for pricing.
Jcorporate
E-mail: info@Jcorporate.com
Web: www.javacorporate.com

JetChart Library

JetChart Library is a suite of Java classes that displays different types of charts, from applets to Java applications and servlets, and in the latter case with server-side .gif file generation. The program's features include dragging and size adjustment capabilities, vertical and horizontal orientation, gradient colors, multiple tooltips display, 2-D/3-D presentation, and support for positive and negative numbers. JetChart supports JDK 1.1 and higher, and is compatible with Internet Explorer 4.x and Netscape 4.06 and higher. Starts at $160.
Jinsight
Phone: +55 82 326 2869
E-mail: mail@jinsight.com
Web: www.jinsight.com

Developer Tools

Flux 1.1

This Java software component performs sophisticated job and task scheduling for Java and Java 2 Enterprise Edition (J2EE) applications. Version 1.1 features improved Cron compatibility and functionality, allowing the program to schedule jobs in a similar manner to Unix Cron. It can integrate tightly with J2EE applications, including Enterprise JavaBeans and JMS messages. Designed to run equally well in server environments and standalone applications, it has been tested against WebLogic, Oracle, DB2, Hypersonic SQL, and Orion Application Server. Starts at $1,000.
Sims Computing, Inc.
Phone: 406-656-7398
E-mail: info@simscomputing.com
Web: www.simscomputing.com

FormulaOne for Java 8.0

FormulaOne for Java 8.0 is a tool enabling developers to dynamically generate and distribute richly formatted Excel spreadsheet reports (not simple CSV or tab delimited files) from databases, XML files, and other data types regardless of a server's operating system and without reliance on the Excel desktop application. It performs scalable calculations, builds reporting and analysis processing, and distributes present data through a multitude of data formatting options. The program also makes it possible to deliver reports inside or outside the firewall as richly formatted Excel .xls files, Formula One .vts files, HTML, or Web-ready images. $399.
Tidestone Technologies, Inc.
Phone: 913-851-2200; 800-884-8665
E-mail: sales@tidestone.com
Web: www.tidestone.com

Java Servlets Office Suite 1.75

Java Servlets Office Suite 1.75 is composed of more than 30 "out-of-the-box" components for Java servlets and JavaServer Pages based on the J2EE. These components include Message Board, Chat, File Manager, and Calendar. Collaboration tools include co-browsing and documents sharing, Portals maker, and more. The tool requires a servletrunner on the server where the site is intended to run. All servlets are configurable and can be embedded into internal design framesets. $399.
ColdJava
Phone: +7 095 151 6159
E-mail: coldjava@usa.net
Web: coldjava.hypermart.net

JBuilder 4 Enterprise

JBuilder 4 Enterprise is a set of visual development tools for developing 100% Pure Java end-to-end applications on the Java 2 Enterprise platform. J2EE e-business applications can be developed and deployed to multiple application servers, including WebLogic and the integrated Inprise Application Server. Java 2 applications can be delivered with Enterprise JavaBeans using JBuilder's Entity Bean Modeler and remote debugging, and business logic connects using InternetBeans Express components, wizards and tools for data-driven Web applications, servlets, and JavaServer Pages. The program also includes a JDataStore 100% Pure Java embeddable database for high-performance data storage. $2,999.
Inprise Corp. (formerly Borland International)
Phone: 831-431-1000
E-mail: customer-service@borland.com
Web: www.inprise.com

JVMs

Jbed RTOS Package 1.3

Designed for embedded systems and Internet appliances, Jbed RTOS Package 1.3 is a combination Java virtual machine (JVM) and real-time operating system (RTOS). Based on 100% Pure Java technology, it allows developers to write entire applications, including device drivers, in Java. The RTOS package runs directly on the hardware, so there is no need for any additional software. The program is available on ARM, PowerPC, 68xxx, and Coldfire processors. Version 1.3 incorporates enhancements in the Jbed TCP/IP networking performance and additional protocols such as PPP, DNS, and daytime protocol. The target bytecode compiler (TBCC) for dynamic class loading also supports loading across HTTP and TFTP. Contact for pricing.
esmertec, inc.
Phone: +41 1 823 89 00
E-mail: info@esmertec.com
Web: www.esmertec.com

Services

PowerUpdate

PowerUpdate is a 100% Pure Java, Internet-based updating service for software applications and data. It alerts customers when there is a product update or newer version of their software available, and then optionally initiates the download and installation of the latest version. Built-in intelligence allows the automatic updating of individual files, multiple files, or the deployment of new installers. PowerUpdate works on any existing installation architecture, including Zero G's InstallAnywhere and in-house scripts. Additional features include an advanced GUI design, HTTP protocol support, the ability to insert update messages, and the ability to manage and update multiple applications from one Web-interface. Free.
Zero G Software Inc.
Phone: 415-512-7771
E-mail: info@ZeroG.com
Web: www.ZeroG.com

Testing and Debugging

Jtest for Linux

Jtest for Linux is a class-testing tool for Java that automatically performs static analysis, white-box testing, black-box testing, and regression testing. Jtest works on any Java classes, including those that use CORBA and Enterprise JavaBeans. $3,495.
ParaSoft Corporation
Phone: 626-305-0041; 888-305-0041
E-mail: info@parasoft.com
Web: www.parasoft.com

SilkPilot 2.1

SilkPilot 2.1 is a tool based on Enterprise JavaBeans (EJB), Java/RMI, and CORBA standards and technologies. It allows developers to automate the functional verification of applications based on complex middleware technologies. The program features an interactive graphical user interface representation of your objects' interfaces, mechanisms for creating regression tests, and a test console with an interface browser, an object window for each target object, and an inventory of objects and data structures. Contact for pricing.
Segue Software Inc.
Phone: 781-402-1000; 800-287-1329
E-mail: info@segue.com
Web: www.segue.com

DOES YOUR JAVA APP SERVER FIT YOUR NEEDS?

When you're developing, you don't need big, bulky functionality that's going to get in the way of what you're trying to do. You need the right tool for the right job. That's the beauty of Allaire™ JRun™ 3.0. Everything you want–clean design, modular architecture, the latest J2EE specifications–to gain greater speed, productivity, and flexibility developing and deploying server-side Java™ applications. And because it's easy to install, you'll be up and running in no time. Which means a perfect fit for your needs, not to mention a solution that makes your development...well, unparalleled.

JRun offers you:

- Ease-of-use
- Reliability
- Scalability
- Multi-platform support

And, new in JRun 3.0:

- Improved developer productivity
- Load-balancing and fail-over
- Enhanced performance
- J2EE support including EJB

allaire JRun™
java application server

<allaire>

JRun, the JRun logo, Allaire, and the Allaire logo are trademarks of Allaire Corporation. Java is a trademark of Sun Microsystems, Inc. ©2000 Allaire Corporation. All rights reserved.

Download your free Developer edition of JRun 3.0 at www.allaire.com/download or call 1-888-939-2545.

Putting up a Good Front

JSP tag libraries allow you to use powerful Java code in your Web site that is cleverly disguised as HTML

by Kevin Jones

JavaServer Pages (JSP), like Active Server Pages (ASP) before it, is an excellent technology for developing dynamic Web pages. It allows the developer to create a page with static and dynamic content. The static content is in the form of (D)HTML, and the dynamic content is written as script. In the case of ASP, you could use one of many script languages, but the most common is JavaScript. In JSP the "script" language is, in fact, Java. Being able to mix dynamic and static content on a page is both a major benefit of these systems and also a major drawback.

It is a drawback for several reasons. First, developing and debugging server pages can be difficult. The page can (and often does) end up being an untidy mixture of HTML and script, which is difficult to read and understand. Second, what if developers want to reuse sections of your script? They have to resort to cut-and-paste reuse, which is not good. And finally, the people that write and design Web pages are usually graphic designers. These folks probably don't know the scripting language used on the page, and learning it may be difficult for them. This is especially true in the JSP world where the language is as powerful and flexible as Java.

Kevin Jones is a UK-based instructor for Developmentor, a training firm based in Torrance, California. A developer with more than 15 years of experience, he has spent the last four years researching and teaching Java programming and most recently investigating HTTP and XML. Kevin can be reached at kevinj@develop.com.

Listing 1 A Library in a Jar This Web page has two tag libraries that are packaged as .jar files: dataaccess.jar and displaytable.jar. When the server encounters a tag that is part of the tag library (in this case, "connection" and "query"), it executes the code in the library.

```
<%- Some JSP Page ->
<%@ taglib uri="/dataaccess.jar" prefix="da" %>
<%@ taglib uri="/displaytable.jar" prefix="display" %>

<HTML>
<BODY>
<HR>
<da:connection driverName="sun.jdbc.odbc.JdbcOdbcDriver"
  sourceName="jdbc:odbc:pubs" >
<da:query id="rs" queryString="SELECT * FROM authors" >
<display:initialiseTable resultsetName="rs">
<display:tableBody />
</display:initialiseTable>

</da:query>
</da:connection>
```

Listing 2 A Great Support The TagSupport class is a helper class. It includes a common implementation of the tag interface.

```
// TagSupport
// methods from Tag
static Tag findAncestorWithClass(
  Tag from, java.lang.Class klass){}

int doEndTag()
int doStartTag()
Tag getParent()
void setParent(Tag t)
void setPageContext(PageContext pageContext)
void release()

java.lang.String getId()
void setId(java.lang.String id)

java.lang.Object getValue(java.lang.String k)
void setValue(java.lang.String k, java.lang.Object o)
java.util.Enumeration getValues()
void removeValue(java.lang.String k)
```

To get around these problems, JSP designers came up with the idea of tag libraries. A tag library is an encapsulation of a behavior that can be reused on many pages. To the tag library author, a tag library is Java, but to the page author, a tag library looks and behaves like safe, friendly HTML. Page authors are already used to HTML, so using tag libraries requires little extra learning. Let's see how they work.

Exploring the Libraries

Listing 1 shows a page that contains two tag libraries, dataaccess.jar and displaytable.jar. Tag libraries are packaged as .jar files that are made available to the Web server. These .jar files contain all tag library classes and maybe even the tag library descriptor (more on that later). When you are building the library, creating .jar files is easy; just use the .jar command line tool. But make sure that full directory information is included in the .jar.

Each library is associated with a prefix, which is basically a short name for the tag library. (If you are familiar with XML, the prefix is used in a similar way to XML namespace. In the next release of JSP, the prefix probably will be an XML namespace). To reference a tag on the page, an HTML/XML-like construct is used. For example, to reference the dataaccess tag library, the listing uses:

```
<da:connection
  driverName="sun.jdbc.odbc.JdbcOdbcDriver"
  sourceName="jdbc:odbc:pubs" >
```

where "da" is the prefix, "connection" is the name of the tag, and driverName and sourceName are attributes of the connection tag. In the listing, "query" is another tag that is part of the dataaccess library. This tag has attributes named id and queryString.

When the server encounters a tag that is part of the tag library, it has to execute the code in the library. The dataaccess library has (at least) two tags. When the server meets the first tag (connection), it searches the library for a Java class that is associated with the connection tag (you will see how that association is made later). When it finds that class, it creates an instance and initializes it. Once the initialization is complete, the tag can do its work.

There are several important things to notice about this example. Tags look a lot like XML/HTML; they have an element name and they may have attributes. A tag library may contain more than one tag. Tags can be nested; for example, there is a parent/child relationship between connection and query in Listing 1. Further, a child doesn't need to be in the same library as its parent. And finally, although this isn't clear from the listing, tags may also process any data that makes up their body. Take, for example, the following code:

```
<pref:someTag>
  This is data for the tag
</pref:someTag>
```

The someTag tag could optionally be passed the text "This is data for the tag" and process it. It might return entirely different text or it might just add HTML markup to it.

Tags will typically generate content, but the tag author determines how this content is generated. For example, tags could open database connections and run queries. Tags could access EJBs and execute methods. Tags could transform XML using XSLT. The possibilities are endless.

Two types of tags exist: tags that don't process their bodies and tags that do

Tag Team

Two types of tags exist: tags that don't process their bodies ("simple" tags) and tags that do ("body" tags). Let's look at simple tags.

Tags have their own library package, javax.servlet.jsp.tagext, that contains an interface called Tag and a class called TagSupport. To enable Tag authoring, Sun provides a library that contains some interface definitions and helper classes. This library is packaged as servlet.jar

Listing 3 Get Started The doStartTag() method is the workhorse of our transformation application. It calls the transform() helper method to transform the XML.

```
public int doStartTag()
{
  java.io.StringWriter writer = new java.io.StringWriter();
  try
  {
    JspWriter out = pageContext.getOut();
    transform(writer);
    out.write(writer.toString());
  }
  catch(Exception e)
  {
    e.printStackTrace();
  }

  return EVAL_BODY_TAG;
}
```

Listing 4 What Have We Here? Our Tag-Library Descriptor (TLD), which looks something like this, reveals important information about our tag library.

```
<?xml version="1.0" ?>

<!DOCTYPE taglib
  PUBLIC "-//Sun Microsystems, Inc.//
  DTD JSP Tag Library 1.1//EN"
  "http://java.sun.com/j2ee/dtds/web-
  jsptaglibrary_1_1.dtd">

<taglib>
  <tlibversion>1.0</tlibversion>
  <shortname>XSLTTransforms</shortname>

  <tag>
  <name>SimpleTransform</name>

  <tagclass>com.develop.kevinj.taglib.Transform
    </tagclass>
  <bodycontent>empty</bodycontent>

    <attribute>
      <name>xslURL</name>
      <required>false</required>
    </attribute>

    <attribute>
      <name>xmlURL</name>
      <required>true</required>
    </attribute>

  </tag>

</taglib>
```

and is available from http://java.sun.com/products/servlet/download.html.

The tag interface looks like this:

```
// tag interface
int doStartTag()
int doEndTag()

Tag getParent()
void setParent(Tag t)

void release()
void setPageContext(PageContext pc)
```

You must implement this interface to write a simple tag. Many tag authors will implement parts of this interface in exactly the same way, so Sun has also provided a helper class called TagSupport (see Listing 2). TagSupport implements all the methods from Tag and adds a few more of its own. To implement a simple tag, typically you would extend TagSupport, overriding the necessary methods.

Building a Transformer

To illustrate tag libraries, we will build a tag that transforms XML into HTML using an XSL transform. The tag will look like this:

```
<%@ taglib uri="/TransformTag.jar"
  prefix="tform" %>
Some HTML
<tform:SimpleTransform
  xslURL="somefile.xsl"
  xmlURL="somefile.xml"/>
Some more HTML
```

This tag will be passed two URLs, one identifying the XML and one identifying the transform (the XSLT). The tag will be a simple tag that transforms the XML using XSLT and outputs the results of that transform.

To write this simple tag, first we have to extend the TagSupport interface:

```
public class Transform extends
  TagSupport
{
  String xslURL = null;
  String xmlURL = null;
```

This class has two data members to hold the URLs.

When the server parses the JSP page and sees the tag, it creates an instance of our Transform class and then tries to set its attributes. The tag has two attributes, xslURL and xmlURL. The container matches the tag attributes to method names using the JavaBeans naming conventions. So, the class can support xslURL and xmlURL attributes with these methods: setXslURL() and setXmlURL(). The methods are simple:

```
public void setXslURL(String url)
{
  xslURL = url;
}
public void setXmlURL(String url)
{
  xmlURL = url;
}
```

Finally the server engine tells the tag to do its work. Because this is a simple tag, all the work is done in the doStartTag() method (see Listing 3). doStartTag() calls a helper function, transform(), to transform the XML. The result of this transformation is held in the "writer" variable. TagSupport has a variable (pageContext) that can be used to access any of the intrinsics used on the page that contain this tag. doStartTag() uses the pageContext to reference the JspWriter. It then calls out.write() to write the result of the transform back to the caller.

Those are the basics of writing a simple tag. Now let's go over some details about how these tag libraries fit into the big picture.

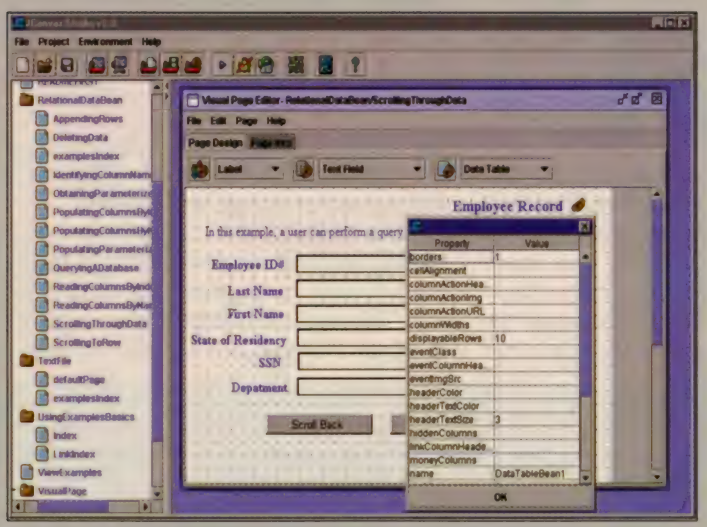
The Translation Phase

JavaServer Pages are compiled into servlets, and the servlets are then executed. When a JSP page that contains a tag or tags is executed, the compiled servlet has to call the appropriate methods on the tag instance at the right time. These methods make up the tag's lifecycle. Part of that lifecycle is (obviously) creating the tag instance. We saw previously, however, that the name of the tag is not necessarily the same as the name of the Java class that will be instantiated. So how does the server know which instance to create?

All tag libraries are self-describing—they contain a tag library descriptor or TLD, which is an XML-format file with a .tld extension that de-

Did You Know?

The best J2EE Development Environment is free?



- Create EJBs, Servlets, and JSPs with ease
- Visual Event/Object Oriented 4GL JSP Development
- Visual JDBC JavaBean Creation
- Integrated Lightweight JSP Container
- Drag and Drop, WYSIWYG component design

Download JCanvas Studio at www.artisticsystems.com

ARTISTIC SYSTEMS™

ADVANCED WEB SERVICES AND TOOLS

7031 Albert Pick Road • Suite 100 • Greensboro, NC 27409 • Telephone (336) 668-4106 • FAX (336) 662-8415

scribes the tag library. The descriptor contains information such as the name of the tag that should be used on the JSP page, the name of the Java class that should be instantiated, the attributes of tag, and whether these attributes are required or optional. If the .tld file is included in the jar file, it must be in a META-INF directory inside the .jar. The .tld file only needs to be included in the .jar file if the URI attribute in the taglib directive is the name of the .jar file used.

JSP architects introduced tag libraries to allow further separation of content from logic

For our tag, the descriptor may look like Listing 4. The document type definition (DTD) for the descriptor can be found at http://java.sun.com/j2ee/dtds/web-jsptaglibrary_1_1.dtd.

So what does this TLD say? First, it says that this is version 1.0 of the library and that the short name of the library is XSLTTransforms (this name would be used by page authoring tools). The TLD then describes each tag the library contains (so far, only one).

For this tag, the TLD tells us the name used on the page (SimpleTransform), and that the associated java class is com.develop.kevinj.taglib.Transform. It tells us that the tag doesn't have a body and that it has two attributes. The first attribute is called xslURL and is optional; the second attribute is required and is called xmlURL. (We make the xslURL optional because the XML document could, itself, reference an XSLT stylesheet with an <?xml-stylesheet ?> processing instruction.)

So how does all this tie together? The server does lots of work when it compiles the page. When the server comes across a tag, it identifies the TLD to be used by matching the tag prefix with the "taglib" directive on the page. The server then checks that the TLD contains a descriptor for the tag with that name.

```
<%@ taglib uri=
  "/TransformTag.jar" prefix="tform" %>
<tform:SimpleTransform xslURL=
  "..." xmlURL="..."/>
```

So in our case it looks for a <tag><name> element that says SimpleTransform. If it finds this entry, the server checks that the .jar file contains the correct class. The server can then check if the TLD specifies whether the tag has any attributes. It looks for required and optional attributes and checks that these attributes have entries in the TLD. If our tag passes these tests, then the page will compile. If any of these steps fail, the page will not compile, and an error will be sent back to the client.

The Execution Phase

When a request is sent to the page, the tag instance has to be created and initialized. The appropriate code was inserted into the servlet at translation time, so no further lookup is necessary at this time.

The Tag interface defines two methods that get called during initialization, setPageContext() and setParent(). These methods are implemented by TagSupport and are called in that order. setPageContext() is passed the current PageContext reference, and setParent() is passed the current tag's parent, or null if the tag is not parented (as in our case). That code would be the same for all tags.

The tag is now initialized with its attributes. Again, the code for this was generated at translation time. Our tag will have its setXmlURL() and setXslURL() methods called. Notice that TagSupport has a method called setId(); many tags will have an Id attribute, and this method saves us from coding the setter function. The tag is now fully initialized and ready to go.

Next, the server calls doStartTag(). Since we're working with a simple tag, doStartTag() contains all of our business logic. In our case, doStartTag() runs the transform and outputs the result. If the tag wants to output any data, it has to retrieve the output stream to write into. The output stream is stored in the PageContext before doStartTag() is called. Remember that the PageContext is passed into the tag during initialization, and that TagSupport provides a helper function. TagSupport.setPageContext stores the PageContext in a variable called pageContext. To access the output stream in doStartTag(), you would use the following code:

```
JspWriter out = pageContext.getOut();
```

You can then use any of the methods of JSP writer to write output back to the caller. Processing the written output can get very complex.

doEndTag() is called when the processing of this tag has finished. After the tag is finished, it can be reused by another page (or another client). This saves the overhead of instantiating the tag again. However, any instance variables must

Return Code	Returned By	Explanation
EVAL_BODY_INCLUDE	doStartTag()	If the tag has a body, then output it.
EVAL_BODY_TAG	doStartTag()	Allow the tag to process its body (BodyTag only).
SKIP_BODY	doStartTag()	If the tag has a body, then do not output it.
SKIP_PAGE	doEndTag()	Do not process any more of this JSP page. We are done.
EVAL_PAGE	doEndTag()	Carry on processing the page.

Table 1 Return to Sender This table lists possible return codes and what they mean.

Listing 5 Resulting Page This is the generated code for our page. All of the steps that we mention are visible here.

```
com.develop.kevinj.taglib.Transform SimpleTransform_0 =
  new com.develop.kevinj.taglib.Transform();
SimpleTransform_0.setPageContext(pageContext);
SimpleTransform_0.setParent(null);
SimpleTransform_0.setXslURL(
  "webapps/tagtest/weather.xsl");
SimpleTransform_0.setXmlURL(
  "webapps/tagtest/weather.xml");
try
{
  int eval_SimpleTransform_0 =
    SimpleTransform_0.doStartTag();
  if (eval_SimpleTransform_0 != Tag.SKIP_BODY)
    {
      do
      {
        out.write("\r\n<BR>\r\nSomeHTML\r\n<BR>\r\n");
      } while (false);
    }
    if (SimpleTransform_0.doEndTag() ==
      Tag.SKIP_PAGE) return;
}
finally
{
  SimpleTransform_0.release();
}
```

Become a Subscriber NOW!

JAVA Pro™

Yes! ☑ Send me one year (13 issues) of *JAVA Pro* for just $29.95

Name: _____

Address: _____

City/ State/ Zip: _____

Country: _____ E-mail: _____

❑ Bill Me ❑ Payment Enclosed Charge my: ❑ Visa ❑ MC ❑ AMEX

Card Number: _____ Exp. _____

Signature: _____

One year is 12 issues plus 1 special issue. Annual cover price is $77.35. Make checks payable to JavaPro. Please allow 4-6 weeks for delivery of first issue. International subscription must be payable in U.S. Dollars plus postage. Canada/Mexico add $18/year; all other countries add $44/year for air mail. Canadian GST included. Java is a trademark of Sun Microsystems.
11/99

4190

Become a Subscriber NOW!

JAVA Pro™

Yes! ☑ Send me one year (13 issues) of *JAVA Pro* for just $29.95

Name: _____

Address: _____

City/ State/ Zip: _____

Country: _____ E-mail: _____

❑ Bill Me ❑ Payment Enclosed Charge my: ❑ Visa ❑ MC ❑ AMEX

Card Number: _____ Exp. _____

Signature: _____

One year is 12 issues plus 1 special issue. Annual cover price is $77.35. Make checks payable to JavaPro. Please allow 4-6 weeks for delivery of first issue. International subscription must be payable in U.S. Dollars plus postage. Canada/Mexico add $18/year; all other countries add $44/year for air mail. Canadian GST included. Java is a trademark of Sun Microsystems.
11/99

4190

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

JAVA™ Pro
PO BOX 54584
BOULDER CO 80323-4584

Subscribe online at
http://www.java-pro.com

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

JAVA™ Pro
PO BOX 54584
BOULDER CO 80323-4584

Subscribe online at
http://www.java-pro.com

The Hottest Magazine for the Hottest Developers in the Enterprise

If you're a professional developer programming in the hottest language in the fastest growing development market in the universe, then you need the hands-on, code-intensive, product-oriented information that comes only from *Java Pro* magazine. Harness Java's unique capabilities in YOUR enterprise with no-holds-barred editorial content.

Don't miss this special offer! Get 13 information-packed issues—a full year of Java Pro *—for only $27.95. That's a savings of more than 61% off the cover price. Call today!*

Java™ Pro: get beyond the hype

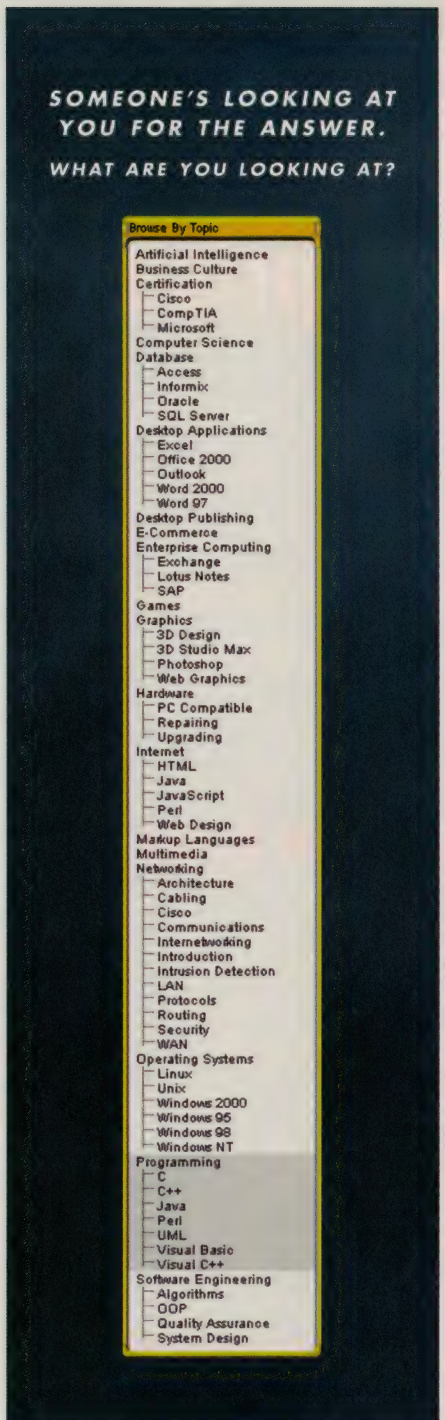
- Practical techniques to amplify Java performance
- Independent, objective product reviews to help you find only the best products for your needs
- Complete source code examples using JavaBeans, servlets and more
- Tips and tricks for Java power programmers

***Java Pro* is the professional developers independent source for building serious Java applications**

Order online today
www.java-pro.com

Subscribe by phone
1-800-848-5523

FTP FAWCETTE TECHNICAL PUBLICATIONS
209 Hamilton Ave.
Palo Alto, CA 94301



There's nothing quite as comforting as seeing the InformIT topic menu appear on your screen. When you see it, you know that a solution is only a few clicks away, and those who look to you, will soon be looking at their hero. www.informit.com

InformIT

eCONTENT EXPERT ACCESS ONLINE TRAINING

be returned to a pristine state. In doEndTag() any instance variables should be reset. So if the page was:

```
<%@ taglib uri=
  "/TransformTag.jar" prefix="tform" %>
<tform:SimpleTransform xslURL=
  "..." xmlURL="..."/>
<tform:SimpleTransform xmlURL="..."/>
```

the same instance could be used for both tags. This is done for efficiency reasons; however this does lead to a slight problem in the current specification. There is no way to re-initialize attributes, so this second tag doesn't "know" that the xslURL attribute is now invalid without doing some processing itself. This issue will be addressed in the next version of the specification. Finally, when the instance will never be used again, the server calls release().

Being able to mix dynamic and static content on a page is both a major benefit and a major drawback

doStartTag() and doEndTag() both have a return type of int, and the values they return affect the rest of the processing of the page (see Table 1). When I mention that the tag has a "body," I mean that the tag looks like this:

```
<%@ taglib uri=
  "/TransformTag.jar" prefix="tform" %>
<tform:SimpleTransform xslURL=
  "..." xmlURL="...">
  HTML here
</tform:SimpleTransform>
```

Since this is a simple tag, doStartTag() could return one of two codes—EVAL_BODY_INCLUDE or SKIP_BODY. A return of EVAL_BODY_INCLUDE would cause "HTML here" to be sent back to the client; a return of SKIP_BODY would not. doEndTag() could return EVAL_PAGE or SKIP_PAGE.

Remember also that the TLD contains a <bodycontent> element. If this is set to empty and the tag has a body, then the page will probably not compile. The generated code for our page looks something like Listing 5.

Server Specifics

The .jar file for this library contains two files, Transform.class (in com/develop/kevinj/taglib) and taglib.tld (in META_INF). The server needs to be able to locate both the .tld file and the class files. In our example, the .jar file was referenced directly from the taglib directive on the JSP, like this:

```
<%@ taglib uri=
  "/TransformTag.jar" prefix="tform" %>
```

If this syntax is used, then the server treats the path to the .jar file as relative to the servlet context. In this case it expects to find Transform.jar in the directory that is the root of the application.

It is also possible to reference the .tld file directly:

```
<%@ taglib uri=
  "/tlds/taglib.tld" prefix="tform" %>
```

In this case, the .jar file can be anywhere on the classpath, but preferably in WEB-INF\lib; otherwise different classloaders get involved and strange runtime errors occur.

The most flexible method is to use a name in the taglib directive and then map that name to a .tld file, using the <taglib> elements in the application's web.xml file. So if you have a taglib directive that looks like this:

```
<%@ taglib uri=
  "/transform" prefix="tform" %>
```

you would need an entry in web.xml like this:

```
<taglib>
  <taglib-uri>/transform</taglib-uri>
  <taglib-location>/tlds/taglib.tld
  </taglib-location>
</taglib>
```

JSP architects introduced tag libraries to allow further separation of content from logic. One of the major aims of tag libraries is to keep script (logic) out of Web pages (content). They allow us to author tag libraries in Java but use them as if they were HTML. The simple tags demonstrated here allow us to run business rules based on attributes and output results back to the client. Make your Web pages easier to organize and manage with these techniques. Your designers will thank you. ■

Go Online

Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0103 Download all of the code for this issue of *Java Pro*.

JP0103KJ Download the code for this article.

JP0103KJ_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

Make Java on the Server More Potent!

AppStream for Java™ Technologies
enables breakthrough performance of Java-based network computing.

www.appstream.com

AppStream™

1-866-AppStream *(toll free)*

©2000 AppStream. AppStream is a trademark of AppStream. All other marks are trademarks of their respective owners.

Squeezing Java into Smartphones

Symbian's David Wood talks about the EPOC operating system and developing software for wireless devices

by Jim Fawcette

With Ericsson's R380, the first cellular phone based on Symbian's EPOC operating system, promised for release in the United States in early 2001 (it's already available in Europe), FTP President Jim Fawcette interviewed Symbian's Executive Vice President David Wood about developing software for smart wireless devices during Symbian's developer conference in London, England. This interview is edited for space and clarity. For the audio of the full interview go to www.xmlmag.com.

Fawcette: *Last July, EPOC 5, the first release of Symbian's operating system, shipped with support for Java 1.1.4. This month (October) we see the first commercial phone based on EPOC—the Ericsson R380—in the market. There are a variety of SDKs and versions. Let's start by bringing our readers up to date on the versions and their timing.*
Wood: We've released four SDKs to date, two each for Crystal and for Quartz. These are our two reference platforms. Crystal is the reference design for a keyboarded wireless device; Quartz is the reference design for a portrait-oriented quarter-VGA size. For each of these devices, we have Java and C++ SDKs. They support Personal Java 1.1.8 and also contain JavaPhone.

In due course, we'll have SDKs for our Pearl smartphone reference design. Next, we'll have releases 6.1 and 6.2 of our OS, which are mainly about enabling packet-based communications, such as GPRS, where the device doesn't have to make a circuit-switched call—the device doesn't have to dial up; it is always connected. With 6.1, we support GSM/GPRS. Release 6.2 does that on a global basis, with support for CDMA and wideband CDMA. There will also be optimizations to improve Java performance.

Fawcette: *Java is not supported on all devices that will ship with EPOC, correct?*
Wood: That's correct. We have a commitment to put Java into every device where it fits the purpose. With the smartphones, that may take a little longer. Not every licensee will include Java with the earliest releases of the smartphones.

Fawcette: *Will that be at the option of the licensees, or will you integrate Java into the various platform OSes?*
Wood: There will come a time when Java is no longer an option for smartphones; it will be built into all of them.

Fawcette: *Before we get into development questions, let's get a sense of the market size we're talking about. If I understand*

David Wood is Symbian's Executive Vice President of Technical Consulting. The Technical Consultancy division provides training, support, technical communications, and technical consultancy to all users of the Symbian platform. David pioneered the use of object-oriented technology within Psion in the late 1980s, was the system architect for the User Interface libraries and application framework libraries for the MC and Series 3 ranges of computers in the early 1990s, and oversaw the architectural development of the Symbian platform. Together with Bill Batchelor, David has recruited and built a world-class software development team that will continue to ensure that the Symbian platform remains the operating system of choice for all major mobile computing and telecommunication device manufacturers. Before joining the Psion Group, David was the head of the math department at a leading London tutorial college. Prior to that, he spent eight years studying math and philosophy of science at Cambridge University (UK).

Jim Fawcette is the president of Fawcette Technical Publications, Inc.

Figure 1 More than a Phone The Ericsson R380 dual-band smartphone is the first commercial phone based on the EPOC operating system. The R380 combines the functions of a mobile phone with a personal organizer, an electronic notepad, and communication features such as e-mail and WAP. Already available in Europe, it will be released in the U.S. in early 2001.

correctly, you're projecting that two million Quartz devices will ship next year.
Wood: That's a conservative estimate, but it depends on when our licensees release their devices. The market is much larger than that for personal computers. By about 2004 or 2005, it is not out of the question to envisage a market of hundreds of millions of devices.

***Fawcette:** What applications would you like to see developed or do you expect will drive the success of this market?*
Wood: That's difficult to answer because we see many of the applications under development by our licensees, but can't discuss them. Some are very compelling. But innovation is always the key. If you look at the Japanese iMode, the applications that are most successful are those that scarcely anyone would have predicted to do so well. They're downloads of cartoons and customizable ringing tones.

***Fawcette:** My understanding is that the R380 is essentially a closed device, so real development begins when the 6.1 OS comes out.*
Wood: That's correct. With the 6.1 OS, applications can be downloaded to the devices. You can write WAP applications for the R380, but what we have in mind starting with 6.1 is more sophisticated applications written in either Java or C++.

***Fawcette:** From your developer's conference, one gets the impression that you are emphasizing Java over C++, is this true?*
Wood: This is the first time that we've released the Java SDKs at the same time as the C++ SDKs, which previously had come out first. But because of the quality of the Java developers we see coming to our Web site, we are making Java a priority.

***Fawcette:** Why did Symbian choose to support the JDK 1.1.x platform, rather than J2ME? Was this an issue of timing?*
Wood: J2ME wasn't around when Symbian started with Java (our first Java release started in January 1997, received Sun's JDK 1.1 certification in August 1998, and was released with EPOC Release 5 in March 1999; Sun came out with the J2SE, ME, and EE categorizations some time later in 1999).

We point out that the Java on Symbian platform v6.0 is "within the J2ME framework," since it uses JDK 1.1 APIs (as required by J2ME), doesn't use Swing (ditto), and uses PersonalJava (an option for J2ME).

***Fawcette:** Correct me if I'm wrong, but I believe your implementation of C++ has keywords specific to your platform to improve error handling and provide the robustness and ability to deal with asynchronous issues, such as dropped calls.*
Wood: C++ is a big language and we strongly advise developers against using certain features of C++ that might make sense on other platforms, but don't make sense on these with their limited memory.

We have our own implementation of exception handling, rather than the standard C++ implementation. For example, C++ allows unconstrained, multiple inheritance. We said that if multiple inheritance is allowed, only one should come from non-mixins and all the others should be from mixins. It turns out that Java does something similar. Actually, we were influenced by the Taligent style guide.

So, we constrain how people use C++. Java makes it much easier. Mobile devices are more demanding environments. The memory constraints mean that you have to think about how you're using resources, but even on a large server, if you run lots of instances simultaneously, you have to manage resources.

***Fawcette:** Why did you choose the guideline "if multiple inheritance is allowed, only one should come from non-mixins?" Is there a performance or resource constraint reason for this?*
Wood: One thing learned from Psion's 16-bit "Sibo" precursor to the 32-bit EPOC system was that it was all too easy even for well-meaning designers to end up with a class hierarchy that had "inheritance encumbrance." That is, designers used inheritance a lot, because it was possible and because it was neat, but in the end it often made the resulting class system harder to understand.

And this was in a system that only supported single inheritance. When multiple inheritance is allowed (as with C++), the risks become even greater.

Figure 2 Easy Communication The Nokia 9210 Communicator is the world's first next-generation mobile phone to be based on the Crystal version 6.0 SDK reference platform. The 9210 Communicator is a dual-band EGSM900/1800 integrated mobile communications terminal that combines phone, fax, e-mail, calendar, imaging, WAP, and Internet access.

We noticed, with interest, Stroustrup's prefatory comments in the multiple inheritance chapter in *Design and Evolution of C++* that boil down to "we did this more for the intellectual challenge than because we thought it was really necessary." Other textbooks in C++ design that we read at the time also seemed to advocate great care with multiple inheritance—lest the complexity of the resulting design explode as it grew in size. So, we decided to restrict multiple inheritance to improve the comprehensibility (and hence the maintainability) of our class hierarchies.

At first, during the early phases of the creation of 32-bit EPOC, we disallowed multiple inheritance completely. However, a couple of the key designers came up with strong cases where multiple inheritance did, in fact, simplify design. We noticed that, in all these cases, only one of the base classes was non-abstract. The Taligent style guide, which we read at about the same time, concurred with this recommendation. So that's what we adopted.

If you look at the Japanese iMode, the applications that are most successful are those that scarcely anyone would have predicted to do so well. They're downloads of cartoons and customizable ringing tones

Over time, similar views have become increasingly adopted by the designers of other object-oriented systems. So much so that, for example, Java enforces, at the language level, exactly what we do in EPOC C++ by convention—namely that a class may extend zero or one base classes, and implement zero, one, or more interfaces. An interface in Java is precisely analogous to a mixin class in EPOC C++.

Fawcette: *What are the tradeoffs between C++ and Java, specific to developing for your platform?*

Wood: There are still things you'd rather do in C++ because of the speed difference. When you've got time loops and computation-intensive applications, you will gain an advantage using C++ instead of Java. The other aspect is that with C++ you can get at the full functionality of the device. There is a huge number of APIs and not all of them are immediately available through Java. We can make up for some of this through JNI, enabling access to another module written to expose these interfaces.

Over time the speed difference between C++ applications and Java applications will become less significant. Hardware will improve; there are some interesting efforts in the next generation of ARM hardware to support Java.

Figure 3 Quartz Platform Ericsson's Communicator products (currently in prototype) are based on the Symbian Quartz reference design, which supports the global wireless network and standards. The Communicator will offer Internet browsing, messaging, imaging, location-based applications and services, mobile telephony, and personal information management. You can download either the Java version or the C++ version of the Quartz 6.0 SDK from www.epocworld.com.

Fawcette: *Symbian has been heavily involved with Java since the 1.1.4 release. What is your role in the reference platform for JavaPhone?*
Wood: We're providing the Java reference implementation for JavaPhone, which makes sense because we know wireless devices.

Fawcette: *What issues emerge when porting Java applications to EPOC; for example, threading?*
Wood: Inside EPOC, we use threads. But our ordinary applications don't always use threads where you would in other operating systems. We have a lightweight alternative to threads called active objects [Editor's note: see sidebar, "Active Objects—What are They?"]. These are visible only to the C++ programmer.

Fawcette: *Will active objects be exposed to Java? (Is that even possible?)*
Wood: It probably isn't possible or even desirable. We provide Java APIs to EPOC services in a Java-like way, rather than a too-overtly EPOC-like way. Simultaneously, we optimize Java thread handling so that, for example,

- EPOC runs Java threads more efficiently
- we implement much more efficient Java ob-

Active Objects—What are They?

The term "active object" refers to an entity in the EPOC operating system that is the building block of an application's event-handling system. In a nutshell, an active object is an object that encapsulates both the code used to determine when a particular event has occurred, as well as the code that is the application's response to the event.

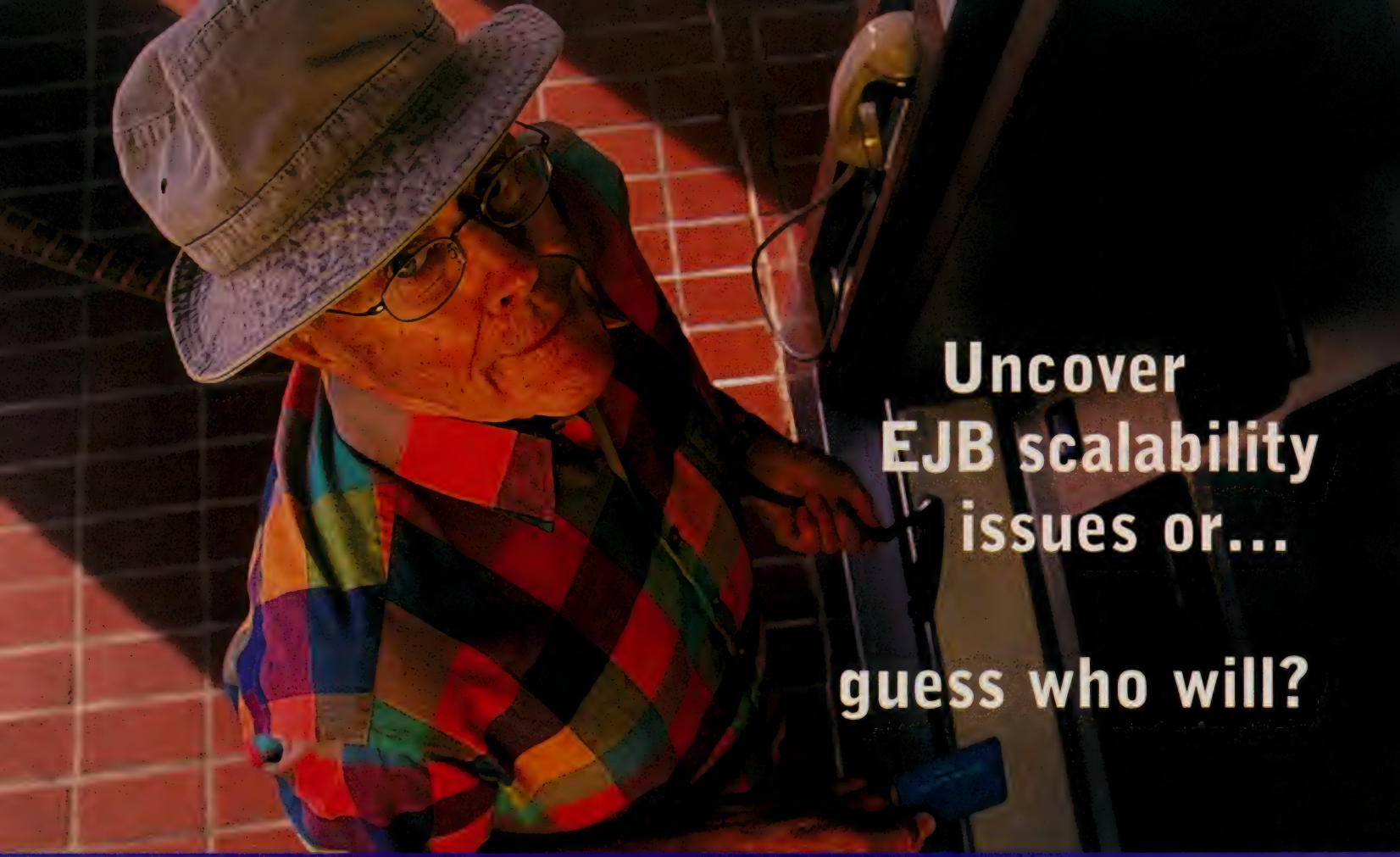
An active object has two phases of operation. When created, functions in the active object that handle its construction will issue a call (usually to a kernel function) that posts a request that the active object be notified when a particular event occurs. In so doing, the active object passes to that kernel function the address of a Status field internal to the active object. The active object then registers itself with the current thread's active scheduler (an entity that handles the scheduling of active objects within the application).

The second phase of the active object's operation takes place when the actual event occurs. When that happens, the kernel code generating the event toggles a bit in the active object's Status field, then signals the requesting thread's request semaphore. This causes the active scheduler to "wake up" and begin scanning its list of active objects, examining each object's status field. The scheduler then launches the event-handling method (RunL(), by name) of the active object whose status field has been toggled.

So, an active object is really a kind of asynchronous event handler. In addition, active objects have priorities that permit the scheduler to decide which active object to "release" first in case of a tie.

It is important to keep in mind that an application in the Symbian/EPOC OS is typically single threaded. The active scheduler is, therefore, a nonpreemptive event-handling loop. No active object—regardless of priority—can interrupt the execution of the currently executing RunL() function.

— by Rick Grehan



Bean-test™ makes sure your EJB™ applications handle the load.

Test the scalability and quality of your EJB middleware applications — *before you go live* — or you may be in for some unpleasant surprises down the road.

Bean-test lets you quickly test your EJB applications. It automatically generates a multi-threaded Java™ test client using your methods. It then populates the client with data and establishes timers for load testing under real-world conditions. Easy-to-interpret graphs and logs let you compare actual results to expected results so you — and your end-users — will never need to worry.

FREE EVALUATION DOWNLOAD!

www.ejbtest.com/ejb3 If you're ready to send your applications out into the world with a lot more confidence, there's no reason to wait another minute. Download a fully functional version of Bean-test today or call 781-993-8500 for a FREE CD.

empirix

STRESSING PERFORMANCE

© 2001 Empirix. Bean-test is a trademark of Empirix. Java and EJB are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are the property of their respective owners.

WWW.EMPIRIX.COM

What is EIKON?

The Symbian/EPOC's GUI is called EIKON (Greek for "appearance"), and it is EIKON that provides the concrete controls with which the user interacts on a Symbian/EPOC-based system. EIKON sits at the top of a rather complex latticework of subsystems that include the window server, the font and bitmap server, and the GDI. (The BTIGDI, which handles rasterising and bitblt operations, is also employed.)

EIKON is also a framework whose classes are the front-line troops, members of which a user application derives and extends. These classes include:

- The application, which serves to define properties of an application
- The document, which represents the data modeled by the application
- The application UI, which handles drawing and screen-based interaction

Another class, the application view—which handles the display of application data and coordinates with the application UI subsystem to deal with user interaction—is part of a lower-level subsystem, but plays a critical role in a user application.

— by Rick Grehan

Resources

Ericsson's R380: www.ericsson.com/r380

Symbian's Developer Program: www.symbiandevnet.com

ject synchronization (happens when two Java threads call synchronized methods in the same object)
- we provide more efficient access from multithreaded Java programs to EPOC services.

Some such enhancements were implemented in GT v6.0. More will follow.

Fawcette: *Will you provide access to the complete EIKON interface from Java applications, or will you provide some sort of Swing-style interface for Java developers on Symbian? (We've seen some wrapper classes that provide access to EIKON components, but it appears that EIKON is only partially exposed to Java developers.) [Editor's note: see sidebar "What is EIKON?"]*
Wood: Symbian has no plans to support or develop Swing.

Over time, Symbian will support ever fuller access to the GUI for a particular reference design. For example, on Quartz, we have provided enough access to write a good-looking Quartz application in Java—which wouldn't have been possible without special classes. We don't yet support *all* features available to C++ programs, but the general direction is to add more support, so that Java programs can aspire to rich native functionality.

One interesting thing is that the Java GUIs are purely graphic-oriented. EPOC's GUIs (along with real-life MFC programming, for example, though we're better at it) also include data and program management, so that our model of how to open, edit, and close a file, how to display embedded documents, how to deal with e-mail attachments or MIME types of Web-based data, and how to launch a program, switch views, link from one app to another, and so on, is closely related to the operation of the GUI. It's in these areas that the integration between Java and EPOC needs the most careful work.

Fawcette: *Do you have any partnerships, or third-party tools we should be aware of?*
Wood: We have relationships with IDE vendors, but I can't discuss those yet.

Fawcette: *What is the role of XML or XML-based RPCs for wireless development?*
Wood: We're very interested in XML and will have support for XML in some future platform, but I can't say if it is 6.2 or beyond. You don't have to wait for us to use XML on EPOC, but there will be system-level support in the future.

We have a commitment to put Java into every device where it fits the purpose

Fawcette: *What does the limitation to support of "outgoing Bluetooth" mean?*
Wood: In 6.0, EPOC devices can initiate a connection, but they can't receive unsolicited connections.

Fawcette: *What is the role of OBEX?*
Wood: OBEX treats data, such as a vCard, as an object. Bluetooth has adopted this as a standard for communicating objects.

Fawcette: *In the corporate market, there are legitimate concerns about supporting wireless devices. If you ask IT managers if they want to support yet another operating system, they'll cringe. Plus, wireless devices raise new levels of security concerns. How is Symbian addressing these?*
Wood: There is extensive support for security in 6.1. If you dip into the SDKs, you'll see a lot there. It would be overkill to have security at the level of Windows NT on a wireless device, but there are numerous opportunities to implement security in EPOC. For example, our database has an interface that enables encryption of data.

There is also a communications SDK, which will be released in January. This has support for the development of custom converters and custom synchronizers. We also have support for custom message types.

Database vendors such as IBM, Oracle, and Sybase support EPOC. The fact that you can get Oracle Lite running on EPOC will help corporate developers migrate to EPOC.

Fawcette: *Although Symbian is a strong supporter of Java and provides the JavaPhone reference, isn't Sun, with Java, a competitor to EPOC? Sun certainly is pushing for handset vendors to implement the JVM as a platform. Given the tight memory constraints, what are the tradeoffs between a full 32-bit OS and a simple JVM for smartphones?*
Wood: There is certainly a category of device where the memory limitations rule out EPOC. However, the kind of Java that can run on these devices is limited—thus Java developers are not exposed to the core wireless functionality of the device.

EPOC provides an efficient Java support environment that is particularly well suited to devices with somewhat larger ROMs and RAMs than the devices targeted by CLDC. The CLDC devices will whet user appetites, and also developer appetites, for more powerful and sophisticated devices. This desire will be well satisfied by Symbian devices.

Fawcette: *Deployment is a growing concern in the corporate market. How will IT deploy applications to widespread, wireless devices?*
Wood: We have a deployment kit, but getting the devices connected is still a challenge. ■

Go Online

Use the following DevX Locator+ code at www.java-pro.com to go directly to this related resource.

JP0103JF_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

HOW MANY INSTALLATIONS DOES IT TAKE TO TARGET YOUR SOFTWARE FOR MULTIPLE PLATFORMS?

Just one. When you use InstallShield's Multi-Platform Editions.

You've probably heard this one before. But this time, it's no joke. We all know that multi-platform projects require unique installation functionality. We're here to say that you can target Solaris™, Linux®, AIX®, OS/400®, OS/2®, and Windows® platforms all at once — effectively creating six installs in one. But only with a true multi-platform solution.

Now you can create a single installation that addresses the unique requirements of each of these platforms AND provides a consistent user experience with a familiar look and feel. InstallShield's Multi-Platform Editions include three powerful product choices to address your specific needs. We deliver a full solution—from support and training to consulting and an upgrade subscription plan—to enhance your multi-platform deployment.

Still think six platforms with one installation sounds like a joke? Guess again.

TRY IT TODAY AT WWW.ONESETUP.COM!

InstallShield
SOFTWARE CORPORATION

© 2001 InstallShield Software Corporation. All rights reserved. InstallShield is a registered trademark and service mark of InstallShield Software Corporation. AIX, AS/400, and OS/2 are registered trademarks of International Business Machines Corp. Linux is a trademark of Linus Torvalds. Windows is a registered trademark of Microsoft Corporation in the U.S. and other countries. All other trademarks are the property of their respective owners.

Networking People

Sprint E-Soutions' networking division needed a faster way to get at legacy project management and personnel data; EJBs with container-managed persistence held the answer

by David Essex and Sue Spielman

For a network-consulting firm like Sprint E-Solutions in Houston, TX, to say that people are your most important asset is no corny platitude. The Sprint division, called Paranet until a 1998 acquisition, and Sprint Enterprise Network Services (SENS) until November 1999, is in the business of sending in teams of people to customer sites to perform networking tasks, even installing entire networks, as it did for Mobil Oil.

In this kind of business, knowing where your people are located, what their skills are, and how much training they need is critical to success. Project management is a key tool, since every job is a project, often with a finite end.

It's not surprising, then, that Sprint E-Solutions field managers spent much of their computing time in PlanView's PlanView project manager tracking billable time and project schedules for the company's 1,500 consultants. They also used Crystal Reports to laboriously extract personnel data such as performance appraisals and pay rates from Oracle and Sybase databases.

The problem was that the reports could take hours to generate. Ten or 15 complaints about the system came in weekly to IS. By mid-1999, marching orders were handed down to director of IS Paul Cali and a two-person programming team: make report generation quicker and easier, using a Web-based interface.

So they built a new three-tier architecture, consisting of legacy databases on the back end, a middle tier of business logic encapsulated in Enterprise JavaBeans (EJBs), and a user interface front end consisting of HTML pages and Java applets (see Figure 1).

The sidebar "In Brief" describes specifics of the Sprint project.

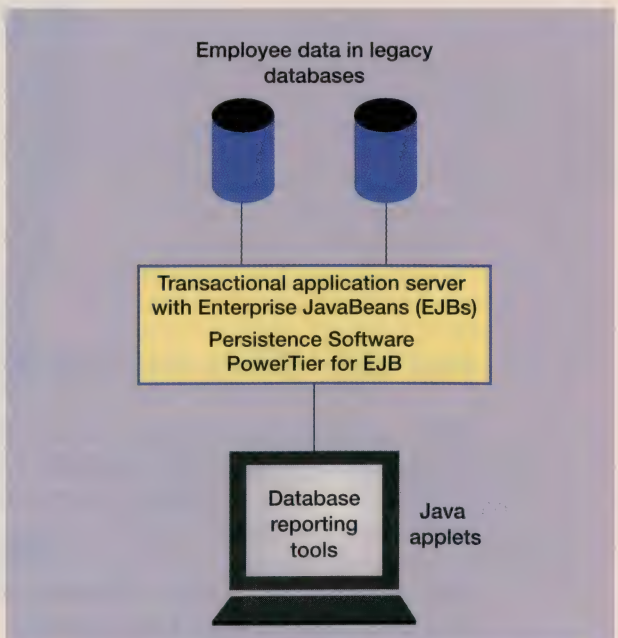

Employee data in legacy databases
Transactional application server with Enterprise JavaBeans (EJBs)
Persistence Software PowerTier for EJB
Database reporting tools
Java applets

Figure 1 A Three-Tier Architecture A new middle tier written in Persistence Software Inc.'s PowerTier for Enterprise JavaBeans has allowed Sprint E-Solutions to quickly develop employee database reporting tools that are faster, more scalable, and easier to use than the legacy system.

App Server Swap

The team knew it needed a transactional application server to handle queries and format them for display in browsers. It quickly settled on BEA's WebLogic Application Server, installing it in August 1999. But by November, WebLogic was scratched and a search for a new app server begun. "With WebLogic you had to make your own entity beans,"

David Essex is a freelance technology writer based in Antrim, NH. He can be reached at david_essex@conknet.com.

Sue Spielman is president and one of the many Java expert consulting engineers of Switchback Software LLC, www.switchbacksoftware.com, specializing in large-scale business and Web application development.

In Brief

The Job
Convert software used to manage consultants who work in Sprint's E-Solutions division. Field office managers had complained that the legacy system was too slow and hard to use.

The Environment
The company opted for Enterprise JavaBeans because of its platform independence and superior stability compared to the Microsoft ActiveX approach, which had proved unreliable in one application. Persistence Software's PowerTier for Enterprise JavaBeans transaction application server was tapped to replace an unsuccessful BEA WebLogic effort, in large part because its container-managed persistence reduced to days the job of coding middle-tier interfaces to corporate databases—normally a two- to three-month process.

The Goals
- Reduce the time spent accessing database queries and reports
- Make reporting tools easier to use
- Gain EJB development experience for possible use in other projects

The Legacy
Oracle and Sybase databases hold information on Sprint's 1,500 on-staff consultants, such as locations, pay rates, resumes, and performance appraisals. Managers use PlanView Inc.'s PlanView to track projects. Under the old system, Crystal Reports was used to access employee performance data. The data sources would all be kept, but the user view would now be unified in a browser interface.

The Solution
- Create a Java-based user interface
- Use PowerTier to build an application server running Enterprise JavaBeans to access Oracle and Sybase data

The Players
Paul Cali, director of IS
Jason Dalton, lead developer
Michael Carlona, developer

The Tools
PowerTier 5.12 (recently upgraded to 6.02), including its Bean Builder Tool; Tek-Tools' Kawa IDE

Hardware
Compaq Computer, Sun Microsystems

Software
PlanView, Crystal Reports (discontinued)

Databases
Oracle
Sybase

recalls lead developer Jason Dalton. Sprint E-Solutions also hadn't been happy with BEA's customer service.

Sprint E-Solutions rejected IBM's application server because it lacked support for the latest version (1.2) of Sun's Java Development Kit (JDK) 2. Likewise, Sun's Web server was eliminated because of poor documentation and an unintuitive programming interface. Finally, Sprint E-Solutions settled on PowerTier 5.12 from Persistence Software Inc.

PowerTier had a huge selling point that won over the team: container-managed persistence, implemented before it became a requirement in the next JDK 2, version 1.3 (and the included EJB 1.1). "It saved us probably a good two to three months of coding," Dalton says. The alternative would have been the much more tedious bean-managed persistence, outlined in version 1.1 of the Java specification, which was made available in December 1999 as part of the first J2EE Final Release. In container-managed persistence, the container—using a mechanism of its choosing—automatically maintains the entity bean data. In bean-managed persistence, the bean is entirely responsible for storing and retrieving its instance data.

"This was possible because of PowerTier's built-in object relational mapping," says Randy Hietter, vice president of market development at Persistence Software. "Developers are not saddled with having to write the low-level JDBC code to interact with the relational database." In one previous and unrelated test, the company showed it could reduce a database interface requiring 450 lines of code down to 26.

EJB containers improve performance by caching not only the beans, but the relationship between the beans, which eliminates the need to access the database layer and execute database-table joins

The team also rejected a Microsoft Active Server Pages (ASP) and ActiveX solution because of bad experiences with a time-entry browser control. "In seven of the last eight weekends, the ActiveX controls had broken down, and people couldn't enter their times," Dalton says. EJB was also favored because of its platform independence.

Dalton and developer Michael Carlona (who is no longer with the company) used Tek-Tools' Kawa integrated development environment (IDE) to write pure Java code that was free of the proprietary code inserted by many IDEs. The Bean Builder tool in PowerTier was also used for some development. Within a few weeks, they built two applications and ran several reports. The total turnaround time for the project was less than two months, according to Cali.

"They had a lot of their business logic [already] done," recalls Kenneth Merkel, a systems engineer at Persistence Software who was involved in the presale, system-modeling phase. "It was a case of writing a back-office infrastructure for it. The biggest thing was that they were running out of time to get their project done."

The most serious programming challenge, according to Dalton, was "getting data out of legacy tables and figuring out how to display them in a way that makes sense." A problem with applet-to-server communication in the old WebLogic-based application also had to be fixed. For an

object to be passed between the two, it needed to be serialized and synchronized, but for some reason one object wasn't, and an inefficient 8 MB Java file was being passed from server to applet. The team solved the problem by creating a new object in PowerTier with both features. The object, which puts a vector (a type of Java list) in a wrapper, facilitates the use of Java Swing applets on Web pages. "It took a bit to get that to work," Merkel says.

Who, What, Where, When, Why

The new Web-based system still incorporates PlanView (which already had browser time entry but not project management), but Crystal Reports is no longer used to generate the personnel-related reports.

A Compaq 1600 server is the production server, while a Dell Enterprise server is used for development; both run Windows NT 4.0. Sun 660 servers run the databases—five for Oracle, one for Sybase, though the Sybase one will be eliminated because of a Sprint deal that makes Oracle products and service less expensive. Offices in 36 cities are linked to each other and Houston headquarters with T1 lines. Some pages are outside a firewall, but most client access must go through Apache authentication and Windows NT RADIUS. Dalton says the company is moving toward using Java security, and he is looking forward to using EJBs' ability to exploit the underlying security of the Web server or operating system to pass information such as a user's name, role, and whether they've been properly authenticated. "It keeps you from having to write a lot of code," he says.

Each application runs on at least one dedicated server along with its related Oracle or Sybase database.

Bean Persistence

The new personnel-management application includes around 250 entity beans and 20 session beans, with 1,400 methods between them. Following the Java model, session beans last as long as a query session, while entity beans "are basically a mirror of the table," Dalton says—the persistent data stored in employee records. One session bean, for example, called GET SDM (Sales Delivery Manager), retrieves Sales Delivery Manager entity beans from databases, according to Dalton.

Servlets and Java Server Pages (JSP) shuttle beans and data between the databases and the GUI. A servlet, for example, might get all the time entered in a given pay period and put it into an object. Then an applet takes the object and displays it in a Swing 1.0 component. Swing components are GUI components and services introduced in JDK 1.2.

Dalton makes liberal use of vectors and Swing's JTables to display database tables in the browser. "We use vectors a lot to put data into a specific object, and then we can persist it to another program," Dalton says. For example, first and last names in an employee database might be represented by an internal vector of table rows that has a second vector of columns wrapped around it. This vector of a vector can then be passed to a JTable statement to create the JTable that the user sees.

Central to all these transactions is the EJB container, which manages enterprise beans and automatically invokes the necessary middleware, including those for lifecycle, state, and transaction management, security, and object persistence. Entity beans delegate their own object persistence to the container, which automatically performs data retrieval and storage operations, saving the developer from having to code those functions within beans. It can also aid performance and scalability by pooling connections, and makes entity beans portable across data stores.

EJB containers also enable PowerTier to improve performance by caching beans in memory. "We cache not only the beans, but the relationship between the beans," Hietter explains, adding that this approach eliminates the need to access the database layer and execute database-table joins. "We can satisfy most inquiries simply by traversing the pre-existing relationships of the bean that are cached in memory," Hietter says. Dalton says the company uses bean caching for nightly database updates, which produces a modest doubling of performance. "The database isn't that big, so we don't see a huge increase," he says.

There is also a failover feature that requires additional, and clustered, servers that can better accommodate "flash crowds" at the site and enhance reliability, in part through cache synchronization done through asynchronous Java Messaging Service (JMS) messaging. Failover has been tested on a server on the development side and will soon be added to a second server. "It seems to work fine," Dalton says.

The object/relational mapping tools in PowerTier radically simplify the mapping that developers must normally hard-code to manage the many complexities among objects. According to a white paper prepared for Persistence Software, "Container-Managed Persistence, Enterprise JavaBeans Persistence Mechanisms," even storing objects in the oft-favored tool, an object relational database, brings its own problems. The "impedance mismatch" between the two types of schema puts an additional burden on programmers to master SQL and JDBC and sort out relationships between objects and data tables. Manual mapping can be error-prone and time-consuming.

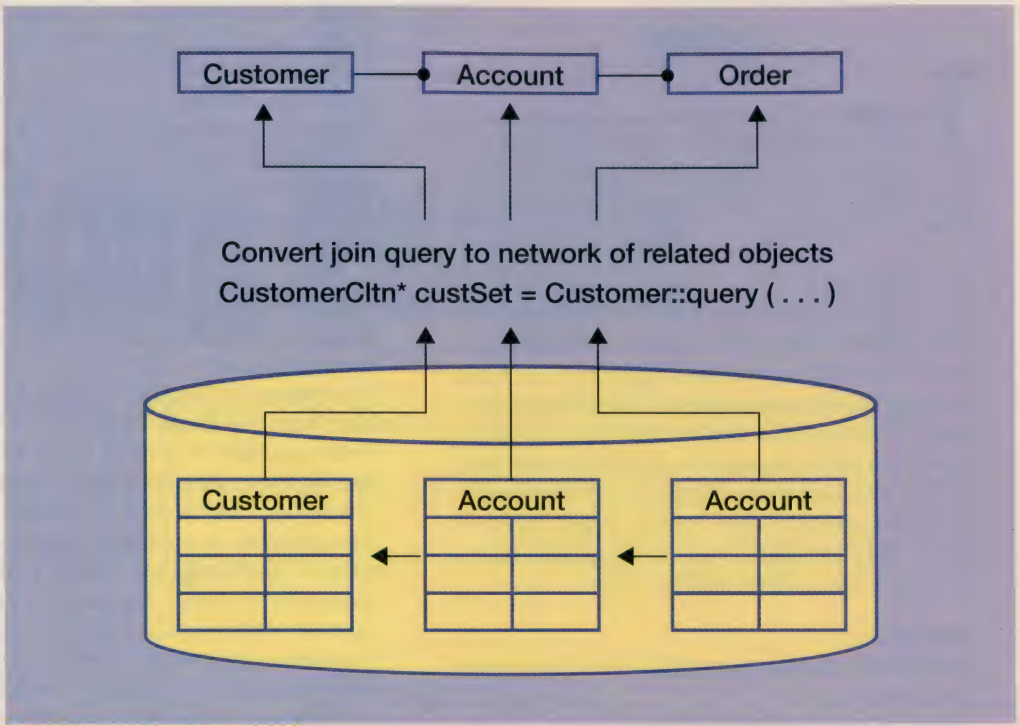


Figure 2 Cache Your Objects PowerTier's ability to automate the complexities of "impedance mismatches" between object and relational databases sealed the deal for Sprint E-Solutions developers, who were working under tight deadlines. One example is executing join queries on cached objects, rather than from the relational database, which can improve performance.

Resources

Persistence Software Inc., maker of PowerTier application server used in Sprint solution: www.persistence.com

Sprint Enterprise Network Services division: www.sprintbiz.com/sens

NEW Scalable Connection Management
Java REALMS Support

Unrelenting Performance

Only FioranoMQ uses a file-based data store to deliver guaranteed messages significantly faster than any other JMS implementation!

FioranoMQ - Message Server

- Fully JMS Compliant, Scalable & Secure
- HTTP Tunneling, XML Interoperability
- Clustering and Failover Protection
- 10x faster than comparable offerings
- Bridges to IBM MQSeries™, MSMQ™
- C++ Interoperability

FioranoMQ™

Internet Messaging Server

FioranoMQ - Info Bus

- Serverless JMS
- Fault Tolerance, High Availability
- Zero Maintenance/Administration
- Reliable Message Delivery over IP Multicast

Fiorano
eBusiness Infrastructure Solutions

Download Free Developer License!

www.fiorano.com, (800) 663-3621, info@fiorano.com

Copyright (c) 2000, Fiorano Software, Inc. All Rights Reserved. Fiorano and FioranoMQ are trademarks or registered trademarks of Fiorano Software, Inc. All other trademarks of their respective owners

In contrast, PowerTier's mapping tools let you use higher-level modeling software to define object models and associate each object to the database schema (see Figure 2). The tools then generate the underlying code. It's this feature that

In container-managed persistence, the container—using a mechanism of its choosing—automatically maintains the entity bean data

Merkel demonstrated to the Sprint E-Solutions team to convince them that PowerTier could help them meet their deadline, and that saved by far the most time in the development effort.

Persistence Software has since released version 6 of PowerTier, notably adding EJB 1.1 compliance and a PowerPage feature in its Object Builder tool that automatically generates JSP pages that have Web access to EJB. Dalton says version 6.02 was given to company developers in September and went into production in October, adding that it enables them to move server information and bean information around without affecting the Web pages of PowerPage. "It keeps things simple," he says. Also, because PowerTier 6.02 creates JSP pages for all tables, end users can more easily delete, modify, or search through the tables than they could using 5.12, according to Dalton.

Reaching out to Customers

Sprint E-Solutions will use the new PowerTier-based infrastructure for other customer-care efforts, according to Persistence Software. Dalton says that means rolling out database report access to Sprint E-Solutions customers by the second quarter of 2001. "What we're envisioning and hoping we can do is provide our customers with a snapshot of what we're billing them for and how much time they have into a project," Dalton says. His team has already created new vacation and utilization matrices for operations managers and is working on a sales force automation application.

Sprint E-Solutions also hopes eventually to allow field managers to do performance appraisals on Palm Pilot handhelds and plans to integrate data from the ResTrack resume program and Computron billing software.

There's little doubt that the team has lived up to its original charge of saving time and improving ease of use. "We've been told that the time sheets we use are giving them back one to two hours a week," Dalton says. The complaints have all but gone away.

From a developer's point of view, the rewards are more esoteric. "We have proved that the EJB structure works," he concludes. ■

Go Online

Use the following DevX Locator+ code at www.java-pro.com to go directly to this related resource.

JP0103DE_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

How to Build a Digital Dashboard
Hands-on Technical Solutions
Exchange & Outlook
rating to
ge 2000
Know Now
Subscribe online
http://exchange.devx.com/in.asp
Call today 1-800-848-5523 and pay only
$39.95 for a full year: twelve issues in all.
One year is based on twelve issues. Annual cover price is $83.40. Please allow four-to-six weeks for delivery of first issue. International subscriptions must be paid in U.S. dollars plus postage. Canada/Mexico, add $18/year; other countries add $44/year for airmail. Canadian GST included.
Exchange & Outlook magazine is another high-quality, how-to technical publication published by Fawcette Technical Publication. Exchange and Outlook are trademarks of Microsoft Corp. All other product names herein are the properties of their respective owners.
Master
Exchange 2000!
With the First and Foremost Exchange Authority
Exchange and Outlook is the ONLY independent Exchange and Outlook magazine, written by the experts exclusively for IT professionals, administrators, managers and corporate developers.
Exchange & Outlook
Migrating to Exchange 2000
Inside Plati
Save 52% off the cover price!

Your Survival Kit for Rapid Deployment of Web Based Applications

Fast JSP/J2EE Development
Easy Administration
Competitive Price

Unify eWave

Visit shop.unify.com
today for FREE downloads
and purchase of
Unify eWave products

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

A Better Way to Build

The Jakarta project's open source Ant build tool provides an easy way to build your Java projects without using makefiles

by Stuart Halloway

Most Java developers I know use an integrated development environment (IDE) to edit their Java programs. As the word "integrated" suggests, IDEs provide most of the different tools you need, all under one roof, including support for editing, debugging, and source control. This combination makes IDEs very effective, but it is not perfect. One problem is that nobody can agree on which IDE to use. On a project with multiple developers, each may choose to use a different IDE. These IDEs do not commonly share information about project structure, so developers must manually replicate project settings to each IDE. This can get so confusing that management may force everyone to standardize on a particular IDE to avoid these problems.

Another problem with IDEs is that they are best suited to interactive development, where a developer can sit in front of a machine and try out new things. Often, IDEs are not well suited to automating routine things such as the daily build or regression test suite. When a process is understood well enough to be automated, developers abandon their IDEs in favor of either shell scripts or makefiles. Shell scripts allow several processes to be launched in a batch mode without human interaction, and typically provide good string-handling capabilities so it's easy to generate the correct command lines. Shell scripts look a lot like other kinds of programs, although they probably will not have debugging features as elaborate as production languages.

Makefiles add to shell scripts the notion of dependencies. Instead of executing an entire sequence of steps in order, makefiles allow each step to be controlled by a dependency rule. Without getting into makefile syntax, a dependency rule allows you to say something like "If a .java file is newer than its corresponding .class file, you need to run javac on that java file." If you use dependencies correctly, your build process can run much faster than if you use a simple script, because you only invoke tools when something has changed, instead of rebuilding the entire project each time. For large projects, this can reduce the rebuild time for a small change from hours to a matter of minutes.

Although you could automate your Java build process with makefiles, this approach has major problems. First, makefile syntax is arcane, unforgiving, and difficult to debug. In fact, it is harder to debug than Java, which suggests that it would be easier to just automate the build process using Java code. Second, makefiles operate by launching a process for each task that is executed. This is necessary when using a heterogeneous set of tools. However, most Java development tools are written in a homogeneous environment—Java. This means that a Java-based build process could invoke the Java tools within the same process, instead of launching a separate process each time. Such a build runs faster because it reduces the number of processes and virtual machines (VMs) that must be created.

Enter the Jakarta Project's Ant build tool. Ant is open source and freely available at http://jakarta.apache.org/ant/index.html. Ant is dependency-rule based, so it has the performance advantage of makefiles. It uses XML as its build file syntax, so it is easy to edit and validate. It is written in Java, so it can connect to the Java build tools directly within a process and leverage the Java APIs. The remainder of this article will get you started managing build projects in Ant, and show you how to extend Ant with your own Java code.

Project Management

To illustrate Ant's basic features, consider a simple Java project that needs to produce source files in a separate directory from output files, a jar file of the .class files, and a jar file of the source.

Following the conventions of the JDK, the project will use these directory names:

```
src - source files
classes - compiled files
lib - jar files
```

To follow along with this example, you will need to download and install Ant from http://jakarta.apache.org/ant/index.html. Then build this directory structure and

Stu is a principal scientist at DevelopMentor (www.develop.com), where he helps develop and teach the Intensive Java, Essential Java, and Guerrilla Java courses. He writes a monthly Tech Tips column for the Java Developer Connection, and he spoke on Dynamic Proxies at JavaOne 2000. When not teaching or writing, he spends most of his free time deep inside the virtual machine. Stu is the author of several tools for Java developers and educators, which can be found at http://staff.develop.com/halloway/JavaTools.html.

Resources

Download the ant binaries, source, and documentation from the ant home page at:

http://jakarta.apache.org/ant/index.html

For help using Ant, subscribe to the Ant user mailing list at:

http://jakarta.apache.org/getinvolved/mail.html. (Be polite by checking the archive http://marc.theaimsgroup.com/?l=ant-user&r=1&w=2 to make sure that your question hasn't already been answered.)

place the source files for one of your projects in the "src" folder. (In order to stay focused on the example, you should pick a project that already compiles correctly and has no external classpath dependencies.)

It is unappealing to have files in the root directory of the project hierarchy, so I am going to keep all the build files in a separate build folder. Ant defaults to use a buildfile named "build.xml," so create a file with that name in the build subdirectory:

```
<!- sample build.xml file ->
<project name="sample"
         default="compile" basedir="..">
  <target name="compile">
    <mkdir dir="classes"/>
    <javac srcdir="src"
          destdir="classes"/>
  </target>
</project>
```

The document element in an Ant build file is the "project" element. The project uses the attribute "default" to specify the name of the default target, so in this example the file will evaluate the "compile" target. Because the build file is in the build subdirectory, the "basedir" attribute value specifies ".." so that all other project instructions can be based from the project root.

Inside the target element is a list of tasks that will be executed in order when the target is evaluated. Ant supports a large number of built-in tasks. In this example, the "mkdir" task guarantees that the "classes" directory exists. Then, the aptly named "javac" task invokes javac to do the actual compile. The javac task specifies a source and destination directory to look for .java and .class files, respectively. The .java file will only be compiled if it is newer than the .class file, or if the .class file does not exist.

To compile your project with Ant, navigate to your build folder and type "ant" from the command line. By default, Ant looks for the build.xml file in the current directory and executes its default task. You should see output something like this:

```
Buildfile: d:\halloway\build\build.xml
compile:
  [mkdir] Created dir:D:\halloway\classes
  [javac] Compiling 1 source file to
          D:\halloway\classes
BUILD SUCCESSFUL Total time: 2 seconds
```

Now, try running Ant again. This time, the classes folder already exists and the java classes are already compiled. So, you should see output like this:

```
compile:
BUILD SUCCESSFUL Total time: 0 seconds
```

As you can see, Ant tasks are smart enough to run only when necessary.

Now that you have the compile task working, edit the build.xml file and add some additional targets, which should appear at the same level as the "compile" target:

```
<target name="clean">
  <delete dir="classes"/>
  <delete dir="lib"/>
</target>
<target name="deploy" depends="compile">
  <mkdir dir="lib" />
  <jar jarfile="lib/projectsrc.jar"
       basedir="."
       includes="src/**,docs/**,build/**" />
  <jar jarfile="lib/project.jar"
       basedir="classes" />
</target>
<target name="all"
        depends="clean,deploy">
</target>
```

Now that the build.xml file contains multiple targets, you can specify the target you want from the command line. First, try the "clean" target by executing "ant clean." The clean target removes all the output files for the project, and is useful if you want to do a build from scratch. After you run the clean target, if you run "ant" or "ant compile" the compile step should run again, since the output files have been deleted.

Although you could automate your Java build process with makefiles, this approach has major problems

Examine the "deploy" target. This target is designed to jar up the source and class files and place them in the lib file. This target is more complex than the other targets so far because it can run correctly only if the compile target has already been evaluated. You specify this dependency via the "depends" attribute of the "target" element. In general, the "depends" attribute will contain a comma-delimited list of other targets that must be evaluated successfully before this target is evaluated. To see this work, return the project to a clean state, using "ant clean." Then, evaluate the "deploy" target with "ant deploy." You should see output similar to this:

```
compile:
 [mkdir] Created dir:D:\halloway\classes
 [javac] Compiling 1 source file to
         D:\halloway\classes
deploy:
 [mkdir] Created dir:D:\halloway\lib
 [jar] Building jar:
       D:\halloway\lib\projects\rc.jar
 [jar] Building jar:
       D:\halloway\lib\project.jar
BUILD SUCCESSFUL Total time: 2 seconds
```

Notice that the "compile" target was automatically evaluated before the "deploy" target. Try deleting various output (.jar and .class) files, and then run "ant deploy" again. You will see that only the necessary tasks run again. Also, try the "all" tar-

Listing 1 Take it to Task One of the benefits of Ant is that you can write your own task in Java. In this case, we have created an Ant task that marks a file read-only.

```
package com.develop.ant;
import org.apache.tools.ant.BuildException;
import org.apache.tools.ant.Task;
import java.io.*;

/**
 * Sets the file named in property "file" to be read-only
 * This capability is already available in Ant using the
 * chmod task, but chmod works on Unix only. */
public class ReadOnly extends Task {
  private String file;
  public void execute() throws BuildException {
    File f = new File(file);
    if (false == f.setReadOnly()) {
      throw new BuildException("File " + f +
        " could not be set read-only");
    }
    System.out.println("Set " + f + " to read-only");
  }
  public void setFile(String file) {
    this.file = file;
  }
}
```

get with "ant all." The "all" target depends on "clean" and then "deploy" so it will build and deploy everything from scratch.

The "javac" and "jar" tasks support a number of additional options that I have not discussed. Both tasks support a standard set of patterns that allows you to match only certain files. The "**" pattern in the "jar" task shown previously is a simple pattern that specifies all files and subdirectories. Ant has a large number of other built-in tasks that cover most of the Java tools, including javadoc, rmic, jarsigner, and keytool. Your installation of Ant includes sufficient documentation of all these tasks to get you started. If the provided tasks don't cover everything you need in your build project, you can also write your own.

Making it on Your Own

One goal of Ant is to bottle the expressive power of makefiles in a clean, cross-platform document structure. Capturing the expressive power is the tricky part, because makefiles are infinitely customizable. Anything you can do from the command line you can do in a makefile. So makefiles can easily handle your organization's custom processes. The problem is that invoking a command-line program imposes only a minimum contract between the make tool and the program. Without more structure, it is easy to write code that is difficult to maintain and debug.

Ant takes a two-pronged approach to extensibility. First, Ant defines a catch-all task called "exec" that can be used to execute a system command. Exec is the Java Native Interface (JNI) of the Ant world, and is little different from a makefile. With exec you can access platform-specific services as you see fit, but you lose the structure provided by Ant. If you use exec sparingly, Ant is still a better choice than a makefile, because most of the build data is stored in a structured format and the unstructured data is easily identified as such. If you overuse exec, your build file will be little more than a makefile hiding inside an XML wrapper. Avoid exec where possible.

The second extensibility option in Ant is that you can write your own task in Java. Adding an Ant task to Java is so easy that you should rarely be tempted to rely on "exec." Listing 1 shows a simple Ant task that marks a file read-only. The ReadOnly task extends the org.apache.tools.ant.Task class, indicating that it can be used as a task by Ant. The ReadOnly task will support only one attribute, named "file," which is the name of the file to mark read-only. For every attribute, the task class should provide a setter method using bean naming conventions—setFile() in this example. The execute() method does the work of the task, and will be called by Ant after all attributes have been set. The execute() method uses org.apache.tools.ant.BuildException to signal failure, or System.out to log success to the build console.

To test the ReadOnly task, you will need to compile it and then make it visible to Ant. When you compile, you will need to set the classpath to include lib/ant.jar from your Ant installation. Once you have compiled the task, add the following lines to your build.xml file from the previous step:

```
<!– place these immediately after the
     "<project..." line –>
<taskdef name="readonly"
 classname="com.develop.ant.ReadOnly"/>
<target name="init">
  <property name="protect-class"
            value="Test"/>
</target>
<target name="protect" depends="init">
  <readonly
  file="..\src\${protect-class}.java"/>
</target>
```

The taskdef element makes the ReadOnly task available to Ant through the alias "readonly." The "protect" target then invokes readonly on your java source file. The only problem is, I don't know what your java source file name is. Instead of hard-coding a file name, you can use a property. The ${...} syntax tells Ant to take the name inside the curly braces and replace it with the value of the property. In order to guarantee that the property exists, you can use a "property" task to assign a default property value, which users can then override from the Ant command line. It is common to group "property" tasks into a separate "init" target and then to make all other targets depend on "init." This guarantees that properties will always have some reasonable value.

To try using the ReadOnly task, you will need to run Ant, specifying a classpath where the ReadOnly class can be found, and overriding the "protect-class" property with the name of one of your .java files. Assuming that ReadOnly.class is in d:/halloway/com/develop/ant and that you have a class named Test.java, the syntax would be

```
set CLASSPATH = d:/halloway/
  ant -Dprotect-class=Test protect
```

If everything works, you should see

```
init:
protect:
  Set ..\src\Test.java to read-only
```

If you get a java.lang.ClassNotFoundException, check to make sure that ReadOnly is actually on your classpath. If you get a "File ... could not be set read-only" message, make sure that you are setting the protect-class property to the name of a .java file in your src folder. ■

Go Online

Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0103 Download all of the code for this issue of *Java Pro*.

JP0103SH Download the code for this article.

JP0103SH_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

TURN XML

INTO EXCEL.

Build dynamic reports on Web and application servers with the Formula One for Java iReporting Engine

Formula One for Java 8.0 attaches to data sources on servers, performs calculations, and dynamically generates and distributes reports in richly formatted Excel spreadsheets, HTML, or images on the Web — all from an API-driven, Excel-compatible reporting engine that you embed in your servlets, EJB, JSP, applets, and Java applications.

iReporting™ for eBusiness

- **Ideal platform for building Java-based reporting applications**
- **Includes internal XSLT processor for filtering and formatting JDBC, XML, and tab delimited data into spreadsheet templates**
- **Dynamically generates Excel reports including formulas, worksheet functions, charts, merged cells, rotated text, outlining, and conditional formatting**
- **Also delivers reports to Web browsers as formatted HTML or through a custom applet that displays live cross-platform spreadsheets**
- **100% Pure Java with no reliance on Windows or Excel**

Attach to data sources.

Filter data into spreadsheet-based report templates. Apply formatting. Perform scalable calculations.

Dynamically generate and distribute reports in richly formatted Excel, HTML, or Web-ready images.

FORMULAONE™
iREPORTING ENGINE
FOR JAVA

WWW.TIDESTONE.COM
800-884-8665

FREE 30-DAY TRIAL

Copyright © 2000 Tidestone Technologies, Inc. All rights reserved. Tidestone, iReporting, and Formula One are trademarks of Tidestone Technologies, Inc. Java, 100% Pure Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the U.S. and other countries. Excel is a trademark of Microsoft Corporation. All other trademarks are property of their respective owners. Specifications subject to change without notice.

Regression Testing at the Unit Level?

by Adam Kolawa

Regression testing is usually considered to be something that is performed at the application level. Not too many people consider performing regression testing at the unit level. Yet this is where regression testing can yield great benefits as a form of error prevention.

Most testing at the unit level, which in C/C++ and Java™ is the testing of a class or function, is performed a few times before the units are integrated into the whole application. To perform unit testing, you need to create test cases for each unit, which involves writing harnesses and stubs for every class.

Though unit testing can be very effective at uncovering errors, taking one more step can make an even bigger difference. Perform regression testing by rerunning all the test cases on the units after each iteration. This will allow you to ensure that no errors have been introduced after any changes have been made to the application.

To perform regression testing, simply save the unit test cases you've already created and gather them into a series of test suites. Once you have the regression test suites you can integrate them into your development process. For instance, you can integrate your regression test suites into the nightly builds that automatically build the application and test it. Your regression test suites are then run with your other tests.

One of the biggest benefits of regression testing at the unit level is that you will detect errors immediately. Any problems can be fixed right after they have been introduced. You'll prevent yourself from having to debug in the later stages of development, when it takes more time and effort to wade through layer upon layer of code in order to fix each error. In addition, these test suites could be used as gates: don't work on other features or move to other iterations until the code passes the test suite cleanly.

By performing regression testing at the unit level, you will gain the benefits of error prevention. You will be ensuring that each unit is solid, that errors won't multiply, and that the application continues to run as expected.

Adam Kolawa, Ph.D., is Chairman and CEO of ParaSoft. You can reach him at ak@parasoft.com

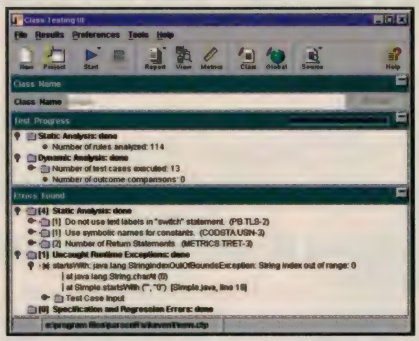
Now Supports Linux!

ParaSoft® jtest®

Click once for automatic white-box testing...and much more

"It is fundamental that every piece of Java code is tested at the class level as soon as it's been written, but writing test cases is tedious and time consuming. Jtest lets me do this automatically and gives me the confidence that my code has been thoroughly tested. I love it!"
—Mansour Raad, DiscoverCase

Jtest® is a fully integrated, easy-to-use, automatic class testing tool for Java™. With the click of a button, Jtest will automatically test any number of classes using the testing procedures explained below.



Jtest is user-friendly; you can start finding bugs automatically at the click of a button.

You do not need to write a single test case, harness, or stub to test a class with Jtest; Jtest does it all automatically.

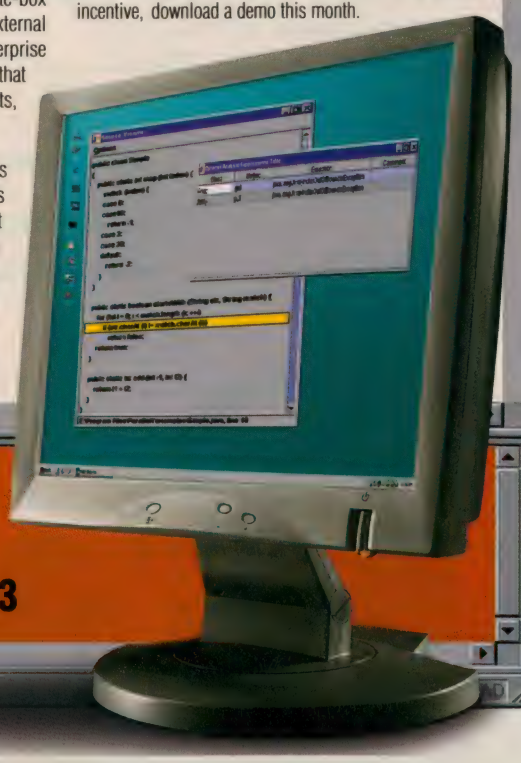
White-box testing prevents crash-causing errors and ensures that classes are robust. Jtest fully automates the white-box testing process by automatically generating and executing test inputs based on the class's structure. Jtest can even generate white-box stubs to test classes using inputs from external resources (such as databases, CORBA, and Enterprise JavaBeans (EJB)). When Jtest finds an input that generates an exception, it reports the arguments, stack trace, and calling sequence.

In black-box testing mode, Jtest checks class functionality using test sets based on the program's specifications. You can provide your own test cases by adding method inputs for simple test cases or creating Test Classes for more complex test cases. If your class references external resources, you can define your own stubs, or you can have Jtest call the actual external method. When you run the test, Jtest uses any stubs you have defined and runs the user-defined inputs along with the automatically-generated inputs. Jtest will then show you the outcomes, which you can verify with one click of a mouse button.

Jtest's regression testing feature lets you perform regression testing at the class level: this means that you can run test suites that monitor your existing code's integrity much earlier than was ever possible before. You can perform regression testing by simply restoring a test and clicking a button, or you can integrate batch-mode Jtest into your nightly builds to ensure that regression errors are always found and fixed as soon as possible.

Enforcing coding standards prevents errors and cuts your debugging time. Jtest's static analysis feature automatically enforces a set of widely-accepted Java coding standards. These coding standards include EJB-specific coding standards and metrics rules which let you control such factors as code complexity. You can also extend and customize Jtest's set of coding standards with the new RuleWizard™ feature that lets you graphically create custom coding standards and modify existing coding standards.

If you would like the satisfaction of using Jtest, visit www.parasoft.com/jpro3, or call ParaSoft toll-free at (888) 305-0041 for more information. For a special incentive, download a demo this month.



ParaSoft®

Download your FREE DEMO at: www.parasoft.com/jpro3

© 2001 ParaSoft Corporation, Monrovia, CA, USA. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the USA and other countries. CORBA is a registered trademark of Object Management Group, Inc. in the USA. ParaSoft and Jtest are registered trademarks of ParaSoft Corporation.

A Time for Reflection

Java 1.2's reflection capabilities eliminate burdensome accept() methods from your Visitor pattern

by Bruce Wallace

In the November 2000 issue of *Java Pro* (Java-tecture), James Cooper surveyed the well-known Visitor pattern. He explained that every element to be visited needs to cooperate by providing an accept() method. This constraint can sometimes be very burdensome. However, a technique is presented in Java 2 that eliminates accept() methods altogether, with only two lines of code, using new reflection capabilities.

The Problem with Acceptance

The Visitor pattern seeks to add functionality to some arbitrary set of classes by factoring out the new functionality into a visitor class. The classes being enhanced by the visitor are known as elements.

In all of the variations of the Visitor pattern, one mechanism or another is used to traverse some set of element objects, visiting each one. However, to do this in the past, each element had to define an accept() method. This method would take a Visitor object as a parameter and call back that Visitor's visit() method, passing itself as a parameter. A typical accept() method would look like this:

```
public void accept(Visitor v)
{
  v.visit(this);
}
```

and a simple traversal code could be:

```
Visitor v = new Visitor();
for (int i=0; i<elementList.length; ++i)
  elementList[i].accept(v);
```

It would be simpler and cleaner if the element classes could remain unaware of the visitations. This is often not just a matter of elegance. When working with external libraries or frameworks, where you don't have control over the source code, accept() methods can't just be added to the classes you would like to visit. Usually this problem is addressed by counseling the developer to create subclasses of these unchangeable element classes. The subclasses can add the needed accept() method. However, this approach still has problems; you usually can't change those same libraries and frameworks to use your new subclass constructors when they instantiate the elements you want to visit.

So, why is accept() used at all? Since most OO languages support what *Design Patterns* calls single-dispatch method calls, and the Visitor pattern needs double-dispatch method calls, the accept() method is the traditional way to simulate a double-dispatch capability. In other words, the version of the visit() method invoked should depend on both the class of the visitor and the class of each element being visited.

However, since the Visitor class must add a version of visit() for each element class that it wishes to visit anyway, why not just call visit() directly? Why not pass visit() the element as follows?

```
for (int i=0; i<elementList.length; ++i)
  v.visit(elementList[i]);
```

While this would work when all the elements are the same class, it won't handle the common situation where elementList holds not simply Element objects, but a mix of objects that are subclasses of Element. Or with the most general case in Java, imagine elementList is an array of Object. The actual class of each element could be anything. In that case, no matter what the real class of each element object was, the method with the visit(Object) signature would be invoked.

With only two lines of code, you can eliminate accept() methods altogether in your Visitor pattern

The Magic of Reflection

In Java 1.2, some new reflection capabilities were added. The new getMethod() call allows you to determine, at runtime, the method of a class that has parameters with specified

Bruce Wallace is president of PolyGlot, Inc., a software development consultancy. He has amassed a quarter century of software engineering experience, and is always keen to find a better way, advance the cause, and do whatever it takes to pay his rent. He can be reached at bruce.wallace@acm.org.

Listing 1 New and Improved Visitor This complete program demonstrates how to define some arbitrary classes and, using reflection, make sure their instances are still visited even though they are collected into a generic Object array.

```
import java.lang.reflect.*;

class Circle
  { public String toString(){ return "I'm a Circle"; } }
class Square
  { public String toString(){ return "I'm a Square"; } }
class Beagle
  { public String toString(){ return "I'm a Beagle"; } }

class MyVisitor
{
  public void visit( Beagle x )
    { System.out.println( "Beagle:[" + x + "]" ); }
  public void visit( Circle x )
    { System.out.println( "Circle:[" + x + "]" ); }
  public void visit( Square x )
    { System.out.println( "Square:[" + x + "]" ); }

  public void visit( Object e ) throws Exception
  {
    Method m = getClass().getMethod
      ( "visit", new Class[]{e.getClass()} );
    m.invoke( this, new Object[]{e} );
  }
}

public class TestBed
{
  public static void main( String[] args )
  {
    try
    {
      Circle c = new Circle();
      Square s = new Square();
      Beagle b = new Beagle();

      Object[] elements = new Object[] { b, c, s };

      // "traverse" the elements visiting each
      MyVisitor v = new MyVisitor();
      for (int i=0; i<elements.length; ++i)
        v.visit( elements[i] );
    }
    catch (Exception x){ System.out.println(x); }
  }
}
```

types. Another new method, invoke(), lets you execute the methods found with getMethod().

A program that uses these new capabilities is shown in Listing 1, which demonstrates a Visitor that doesn't need accept() methods. This program defines three arbitrary element classes (Circle, Square, and Beagle) that we wish to enhance with a Visitor. To keep things brief, MyVisitor doesn't enhance these classes much, but it does follow the Visitor pattern by defining a visit() method for each one. It also adds a generic visit() method that has Object as its argument type. The magic two lines of code in this method use the reflection API to determine the real class of the element parameter. The appropriate visit() method for the element's actual class is found and invoked. Finally, a TestBed main program is defined. It instantiates elements of various classes and collects them into an Object array. It then creates a visitor object and invokes it while traversing the element collection.

This technique can be used with any collection of element classes, and with any type of collection traversal or iteration technique. Also, the traversal can be managed by the element collection, the visitor, or a third party, without restriction. The only condition for using this technique is that you must use Java 2 for your Visitor objects. But since the elements themselves don't need to be modified, they don't have to be Java 2 objects. ■

Go Online

Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0103 Download all of the code for this issue of *Java Pro*.

JP0103BW Download the code for this article.

JP0103BW_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

developmentor
A DEVELOPER SERVICES COMPANY
Helping developers with training, software & publications.

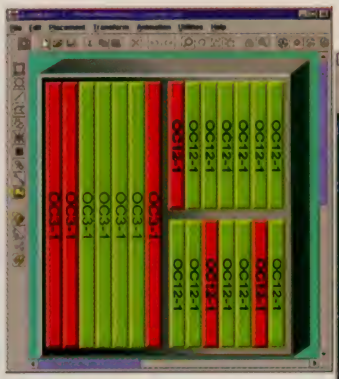
www.develop.com
LOS ANGELES • BOSTON • PORTLAND • LONDON
Schedule of training courses, and other developer resources are available by calling 800.699.1932 or on our web site.
"mind the gap"
Java • JDK • VM • JTS
XML • JNDI • EJB
Security • JSP • J2EE
JAAS • JDBC • JMS
RMI • JNI • JVMPI

Accelerate your Java™ Graphics Development Today

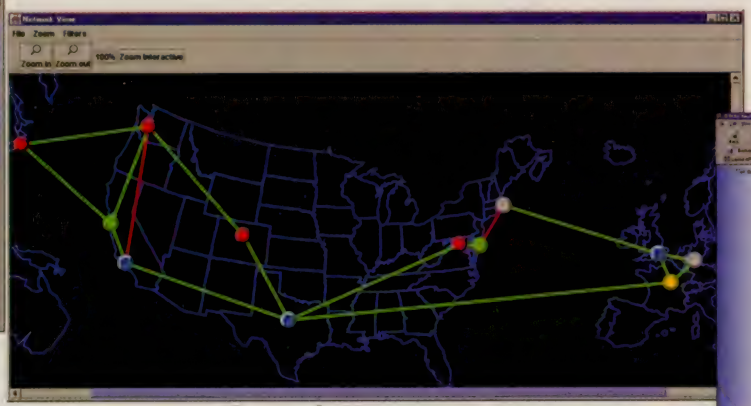
Ask about our free 30-day evaluation program

JLOOX

JLOOX is the best-in-class graphics toolkit for Java 2 developers. Based on Java2D, JLOOX provides a high-level 100% Java API that offers breakthrough performance and productivity gains for developers of graphics intensive Swing applications. It is ideal for applications requiring highly graphical displays such as networking and telecommunications,intelligent transportation systems, process control and simulation.

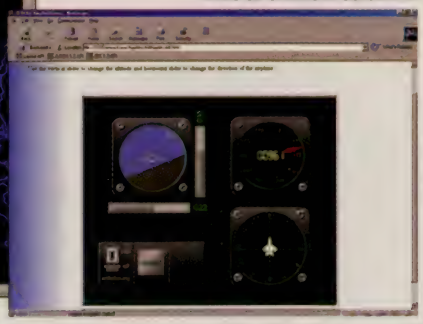


Element Management



Network Management

Instrumentation



Unique advantages:

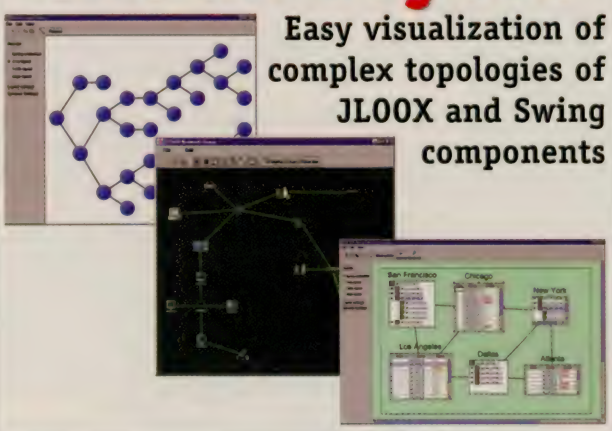
- Includes an interactive editor for rapid creation and animation of graphic screens
- High-level intuitive API decreases development time
- Pre-built collections of objects provide powerful data-visualization primitives
- Offers the rendering power of Java2D
- Includes the award winning JClass Chart from KL Group
- Can be used with any IDE supporting Java 2
- Applications created with JLOOX have no runtime, royalty or deployment fees

JLOOX GIS

Ultra-fast Java mapping components

JLOOX Layout

Easy visualization of complex topologies of JLOOX and Swing components



Accelerating Graphics Development™

LOOX Software Inc.

JAVA PRO 2000 READERS' CHOICE WINNER

1860 El Camino Real, Ste. 306, Burlingame, CA 94010 • T:650-552-9310 F:650-552-9307 • www.loox.com

Copyright (c) LOOX Software Inc. 1999. LOOX is a registered trademark and JLOOX is a trademark of LOOX Software Inc. All other trademarks are the property of their respective owners.

Borland®

Make the jump to JBuilder 4

The sky's the limit with the #1-rated Java™ development tool.

JBuilder™ 4 is the leading cross-platform environment for creating Pure Java end-to-end solutions built on the Java™ 2 Enterprise platform. Bring your e-business applications to market faster with team collaboration, rapid Web development, and application server independence.

Borland® JBuilder 4 will take you to the extreme...

where almost anything goes.

Order at **shop.borland.com**
or call 1-800-252-5547

www.borland.com

Copyright © 2000 Inprise Corporation. All rights reserved. 11672.1

Javatecture

Testing the Limits

Researchers analyze if the knowledge of design patterns helps programmers write better code and communicate their work

BY JAMES W. COOPER

It's just like a political campaign. The candidates get up there and say the same comforting things over and over. Nothing you haven't heard before. *Heard* indeed! Maybe it's just a *herd* mentality. Or maybe as Gilbert and Sullivan said, "The prospect of a lot of dull MPs, in close proximity./ All thinking for themselves is what/ no man can face with equanimity." Does this apply to patterns? Do we all assume they are the magic potion or feather, but without any proof? If patterns are the answer, what is the question?

You've probably heard a lot about design patterns, both from me and from lots of the other Patternostra. Intuitively, they seem to be elegant ways of describing useful reusable designs. But has anyone checked to see if they actually make a difference?

I recently heard a talk by Professor Walter Tichy from the University of Karlsruhe in Germany, who described work with his colleagues Lutz Prechelt, Barbara Unger, and others. They devised and carried out some experiments to find out if the knowledge of design patterns really helps programmers write better code.

The Exam Experiment

They set out to measure whether programmers write better code if they know that design patterns are used than if they do not. To a large degree this is a real Heisenberg problem, because it is so difficult not to perturb the system you want to measure. With that caveat firmly in mind, the researchers set up a final exam situation in which students were given two programming problems to solve. They were given well-commented programs and asked to describe how they would modify the programs. They did not have to write working code, only describe the methods and classes they would add.

In one case, students were told the names of design patterns contained in the base program (in a few additional comment lines), and in the other case they were not. Otherwise, the programs were the same. The students were divided into four groups. Two of the groups were given problem A with the additional comment lines showing the names of the design patterns (call it Ap) and problem B without the design pattern comments. To avoid a learning effect, one group got Ap then B, and the other group B then Ap. The other two groups were given A then Bp or Bp then A.

As background, all of these 74 students had been given an intensive six-week course in Java programming, in which they learned about the Composite, Observer, Strategy, and Template methods, and Visitor patterns, and had assignments in which they practiced using these patterns. The students were uniformly distributed into the four groups by their class grades at that point in the course.

Then, before the exam questions were given, the students were asked in a pretest how well they believed they understood design patterns and in six specific programming cases, which pattern would be most appropriate.

The two exam problems used the Template method, Observer, and Visitor patterns. One was a simple program for reading in and displaying phonebook-style information using a Template method and Observer pattern, and the other a program for walking through a tree of And/Or nodes using a Visitor pattern. The students took two to four hours to complete the exam.

In the case of the relatively simple phonebook program, no significant differences in the number of correct solutions emerged, but the subjects who were told that a pattern existed solved the problem significantly faster. For the harder tree problem, the number of correct solutions was significantly higher in the group told about patterns. Further, pattern knowledge allowed even the less talented students to produce a correct solution.

I think these results are significant and tell us a great deal about the value of patterns. Of course you could argue that since the students didn't write complete, working programs, they weren't doing software engineering, but only simple puzzle solving. To try to measure this same information in a context where working programs were required, the investigators joined Douglas Schmidt at Washington University in St. Louis, MO. Here they performed a similar experiment on a class of 22 students in a C++ programming course. The example programs were thus all written in C++ and commented as before. The students in this course had to implement their solutions on Unix workstations. About the same number of students came up with the correct solution to the And/Or

Resources

Lutz Prechelt's papers:

http://wwwipd.ira.uka.de/~prechelt/Biblio

Download *Do Design Patterns Improve Communication? An Experiment with Pair Design*, Barbara Unger and Walter F. Tichy:

http://wwwipd.ira.uka.de/EIR/wess.pdf

tree (four of eight with design pattern comments and three of eight without). However, the mean time to complete the programs was significantly shorter for the group of students with design pattern comments (52 versus 67 minutes). So again, knowing about design patterns helped.

Before you criticize these experiments as too complex, too simple, too easily perturbed, or too something else, you should read the papers the investigators wrote on these experiments. Take a look at Prechelt's Web site (see Resources). Not only did the researchers subject their data to exhaustive statistical analysis, they thought through all kinds of possible objections to the experimental design. For example, did the students who were not told of the existence of patterns receive enough information? Was the experiment "fair?" To answer this one, the programs were extensively commented as to the algorithms and method level. The only difference was a comment like the following:

```
// **Design Pattern**
// The two displays are registered as
// Observers
```

You might also be concerned that these problems were too simple and not representative of the kinds of problems working programmers encounter. Further, the subjects were students, not experienced professional developers. Might this skew the results in some way?

How the Professionals Did

One way to find this out was to take a set of professional developers and perform, more or less, the same experiment. The investigators studied a set of 30 professional developers of varying experience levels at a single software company. They gave them a pretest, a short course on patterns, and then a posttest. The tests were carried out on paper, without the need to write complete working code.

They found that in the pretest, developers carried out a module (nonpattern) solution faster than a pattern solution. After the course, some of the developers carried out the pattern solution faster than the modular solution, and the speed of the modular solution was unchanged from the pretest. None of the developers carried out the pattern solution more slowly. They concluded that patterns, in general, can help and never hurt.

Now, remember that the name of a thing is what gives you power over it. If you can provide the name for a pattern in your code, you can tell others about the system in a more efficient way. So, is this just a hypothesis, or is it borne out by experience?

To test this question, Barbara Unger carried out an experiment using two-person teams of programmers. She gave one of the programmers a program description and a maintenance task. This first programmer became the "experienced" programmer, who was then to describe the problem to the other "novice" programmer, and the two of them were to carry out the solution.

The experiment compared the teams' interactions both with and without pattern knowledge and studied how they communicated about program design. The teams were given one of the two problems as a pretest. Then they took a three-month lab course on design patterns, followed by the other exercise as a posttest. Some students got problem A before the course and problem B afterwards, and others got B, then A. The expert and novice roles were also switched. One exercise used the Observer, Bridge, and Singleton patterns and the other the Composite, Visitor, Chain of Responsibility, Observer, and Singleton patterns.

Unger measured the communication between the two programmers, looking for questions, use of atomic assertions, and design pattern terminology. She also measured who dominated the discussion and found that, as you would expect, the experienced programmer dominated the discussion at first and continued to dominate the discussion in the pretest experiment. However, in the posttest experiment, when design patterns were now part of their shared vocabulary, the junior team member took a more active role in the discussion, and the time each spent speaking became more equal. This took place regardless of how successful the team was in the proposed programming task. This clearly makes the team's communication much more effective. In addition, in teams where the novice dominated in the pretest, even though the novice was considerably less informed, this domination by ignorance did not take place in the posttest. In fact, the "ignoramus effect" was muted. After the training, the "ignoramus" was no longer ignorant and participated without dominating. This result, too, is powerfully compelling, and one we would all like to see more frequently.

If you can provide the name for a pattern in your code, you can tell others about the system in a more efficient way

This work seems to provide convincing proof that design patterns really do help you write better, faster, more correct code, and help in communicating your work to others. I not only use patterns in my daily programming, I tell others they are there. Try this and see if it helps you, too. ■

James W. Cooper is a computer science researcher and the author of 13 books. His latest, *Java Design Patterns: A Tutorial*, was published by Addison Wesley Longman in January 2000.

Go Online

Use the following DevX Locator+ code at www.java-pro.com to go directly to this related resource.

JP0103JT_T Read this feature online. DevX Premier Club membership is required.

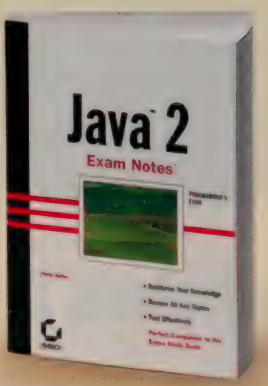
Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

©2000 Amazon.com, Inc. All rights reserved. Amazon.com and Earth's Biggest Selection are trademarks of Amazon.com, Inc. All other trademarks are the property of their respective owners. Amazon.com reserves the right to limit quantities on selected items.

Enough Java books to fill your office and your garage.
(On second thought, they may be one and the same.)

Java In A Nutshell
ISBN: 1565924878

The Complete Java 2 Certification
Study Guide
ISBN: 0782128254



Java 2 Exam Notes
ISBN: 0782128262

Java Server Pages
ISBN: 156592746X

These are four of the thousands of Computers & Internet books available at Amazon.com.

java web development perl web servers xml html javascript wap
visual basic bluetooth ruby site design cryptography **the book store**
c++ web graphics java certification cgi programming jaguar and more

earth's biggest selection.™
amazon.com.

Opportunity Doesn't Always Knock,

Sometimes it Rings.

Java™ wireless development is here.

New wireless devices, such as the latest phone from Motorola, are creating a lucrative market for Java developers. Take advantage of this newest opportunity with CodeWarrior for Java 6.0, your fastest path into the wireless market. Our comprehensive set of tools allows you to compile, emulate and debug Sun Microsystems' standard J2ME™ (CLDC/MIDP)–enabled wireless devices. CodeWarrior Java tools support J2SE™, PersonalJava™ and J2ME open standards, so you get powerful desktop and device development tools, all in one box. Open the door to one of the most promising opportunities in years — with CodeWarrior for Java 6.0 tools. For more information, or to order, log on to **www.metrowerks.com.**

CodeWarrior
Java™ Development Tools

metrowerks
Software Starts Here

©2001 Metrowerks Corp. All rights reserved. Metrowerks, the Metrowerks logo and CodeWarrior are trademarks or registered trademarks of Metrowerks Corp. in the United States and other countries. MOTOROLA, the Stylized M Logo are trademarks of Motorola, Inc. ®Reg. U.S. Pat. & Tm. Office. J2ME, Java, J2SE, Personal Java are trademarks owned by Sun Microsystems. All other products are trademarks of their respective companies and are hereby recognized.

Out of Left Field

By providing a mechanism that lets you add or remove validators, JValidatingField adds flexibility and reusability to your procedures

BY CLAUDE DUGUAY

One of the new features in the Java 1.3 release is a field validating architecture. You can use an InputVerifier to test whether the value in a given text field is appropriate before allowing the user to change the focus. This powerful mechanism lets you use a verifier with any JTextField simply by extending the InputVerifier class and associating it with the field by using the setInputVerifier() method. Unfortunately, developing individual verifiers for each field can quickly become tedious. When I first used this paradigm, it occurred to me that a library of validators that could be associated with an InputVerifier subclass would be more flexible and considerably more reusable.

Figure 1 shows the relationships between classes we'll develop. Of particular note is the large number of classes that implement an interface called FieldValidator. This is a simple interface that exposes two methods. The isValid() method takes a String argument that reflects the value of the field and returns a boolean response, indicating whether the value is valid. The getMessage() method returns a String message that can be presented to users to guide them toward correcting the problem. In each of these implementations, I've supported two constructors: one without arguments that provides a generic message, and one in which you can specify the message more explicitly.

You'll notice an AbstractFieldValidator that exists primarily for convenience. This class implements reusable message management and a method called isNull(), which can be used to avoid null pointer exceptions. The SwingFieldVerifier extends the Swing InputVerifier class and lets you add and remove FieldValidator instances. Whenever the InputVerifier's verify() method is called, each of the FieldValidator instances is checked

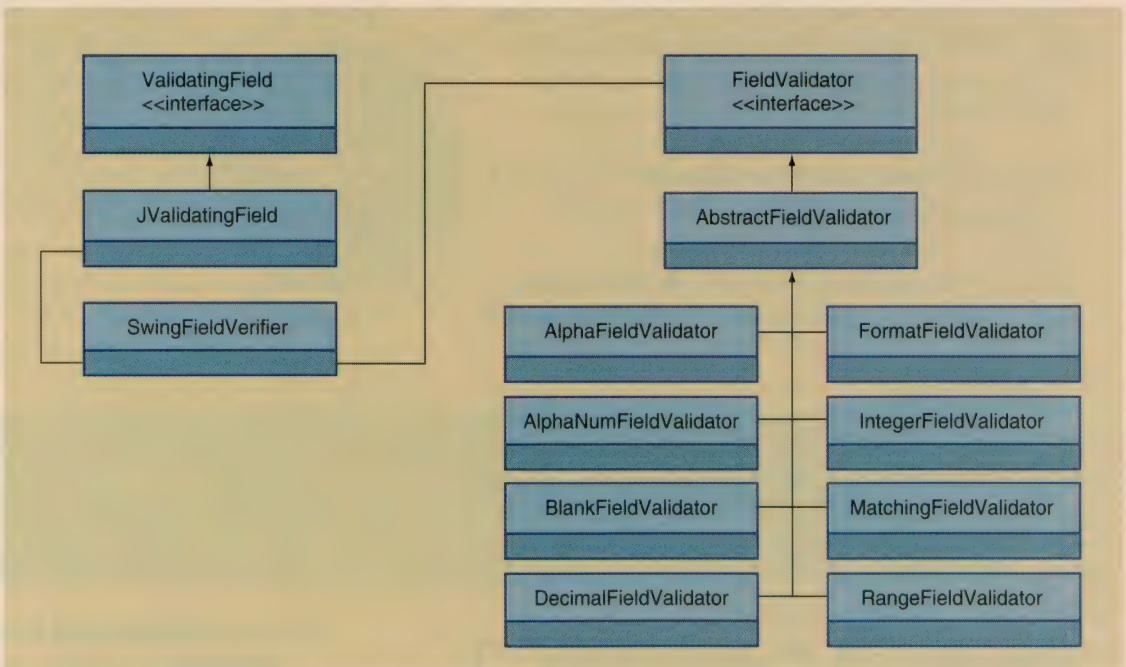


Figure 1 Class Relationships Many classes implement the FieldValidator interface, which exposes two methods: isValid() and getMessage(). The isValid() method takes a String argument that reflects the value of the field and returns a boolean response, indicating whether the value is valid. The getMessage() method returns a String message that can guide users toward correcting the problem.

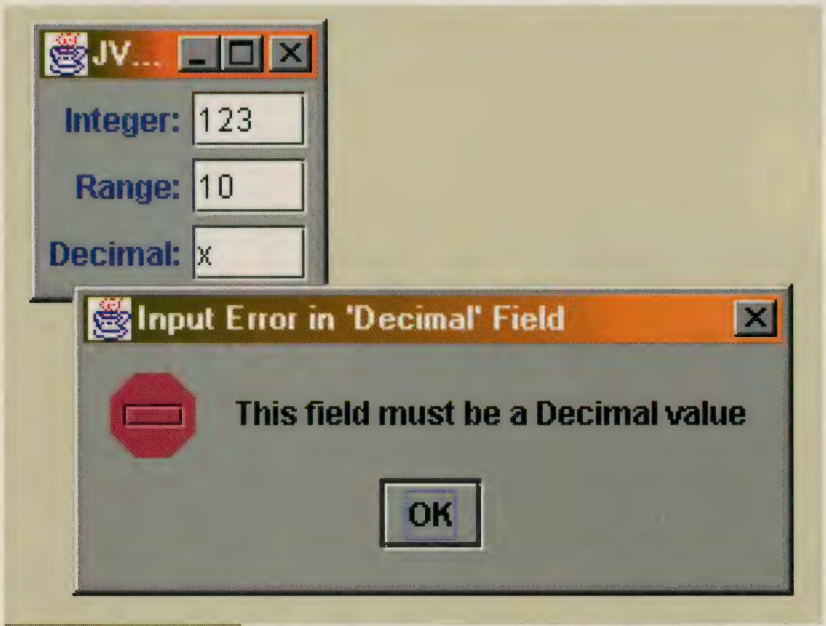


Figure 2 Bad Answer! Each time the InputVerifier's verify() method is called, the FieldValidator instance is checked by calling isValid(). If any responses are negative, the user sees a dialog box with the associated message.

by calling its isValid() method. If any negative responses are encountered, the associated message is presented to the user in a dialog box (see Figure 2).

The code for the SwingFieldVerifier class is shown in Listing 1. The verifier method takes care of the casting you're expected to do when you implement an InputVerifier. I think that the use of generic objects can be dangerous in this interface. Given that only JTextField objects are ever passed as arguments, this may be rela-

Because developing individual verifiers for each field can be tedious, a library of validators associated with an InputVerifier subclass would be more flexible and reusable

Listing 1 Get Flexible We extend the Swing InputVerifier class and enable FieldValidator instances to be added or removed at will, enabling us to verify each, in turn, during the call to the InputVerifier's verify method. This adds considerable flexibility to the design.

```
import java.awt.*;
import java.util.*;
import javax.swing.*;

public class SwingFieldVerifier
  extends InputVerifier
{
  protected String name;
  protected Component parent;
  protected ArrayList validators = new ArrayList();

  public SwingFieldVerifier(Component parent, String name)
  {
    this.parent = parent;
    this.name = name;
  }

  public void addValidator(FieldValidator validator)
  {
    validators.add(validator);
  }

  public void removeValidator(FieldValidator validator)
  {
    validators.remove(validator);
  }

  public boolean verify(JComponent component)
  {
    JTextField field = (JTextField)component;
    return isValid(field.getText());
  }

  public boolean isValid(String value)
  {
    for (int i = 0; i < validators.size(); i++)
    {
      FieldValidator validator =
        (FieldValidator)validators.get(i);
      if (!validator.isValid(value))
      {
        JOptionPane.showMessageDialog(parent,
          validator.getMessage(),
          "Input Error in '" + name + "' Field",
          JOptionPane.ERROR_MESSAGE);
        return false;
      }
    }
    return true;
  }
}
```

Listing 2 Using the Extension This JTextField extension automatically uses the SwingFieldVerifier class and exposes the addValidator() and removeValidator() methods so that it's easy to work with.

```
import java.awt.*;
import javax.swing.*;
import javax.swing.text.*;

public class JValidatingField
  extends JTextField
  implements ValidatingField
{
  protected String name;
  protected SwingFieldVerifier verifier;

  public JValidatingField(String name)
  {
    this(name, new PlainDocument(), "", 0);
  }

  public JValidatingField(String name, int col)
  {
    this(name, new PlainDocument(), "", col);
  }

  public JValidatingField(String name, String text)
```

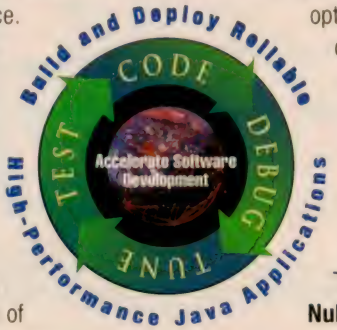
Continued on page 56.

Don't Let One Bad Applet Spoil the Whole Application!

TUNE PERFORMANCE ACROSS YOUR ENTIRE DISTRIBUTED APPLICATION WITH NUMEGA® DEVPARTNER® 2.0 JAVA™ EDITION.

Problems with memory use, multi-threading and slow code can "spoil" your application's performance. **NuMega DevPartner 2.0 Java™ Edition** identifies the bad apples in your distributed, web-enabled and e-business applications—quickly and easily.

DevPartner Java Edition is a suite of *performance-tuning* tools that helps you build reliable, high-performance applications and components with Java technology. With support for distributed application development on a range of application servers, servlet engines and web servers,



DevPartner Java Edition allows you to profile, test and optimize components across all tiers of your distributed application so you can measure and improve your application's performance. And, it is the *only* development productivity suite that extends beyond Java code to support web scripting technologies such as JavaScript®, JavaServer Pages® and Active Server Pages.

Get rid of those bad applets and more! Tune your applications for high-performance with **NuMega DevPartner 2.0 Java Edition.**

NEW! Support for Solaris, Linux, Servlets, EJBs and Application Servers.

Accelerate Software Development with NuMega DevPartner 2.0 Java Edition!

Call **1-800-4-NUMEGA** to request a **FREE** copy of our *"Effective Memory Utilization for Reliable, High-Performance Java Applications"* technical paper. Or download the paper today by visiting us at **www.compuware.com/numega**.

1-800-4-NUMEGA
www.compuware.com/numega
30-Day Money Back Guarantee

COMPUWARE®
People and software for business applications™

Copyright © 2000 Compuware Corporation. All rights reserved. Java and all Java-based marks are trademarks of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are the property of their respective owners. ® indicates registration in the United States.

tively safe. I can see why designers wanted the interface to be generic, but forcing implementations to cast JTextField instances is not exactly an elegant solution. In any case, our implementation does the verification by iterating through the list of FieldVerifier instances in our isValid() method. The only other methods, addValidator() and removeValidator(), allow you to add and remove validators to and from an ArrayList.

The JValidatingField class in Listing 2 is a JTextField extension that automatically uses the SwingFieldVerifier class and exposes addValidator() and removeValidator() methods to make it easy to work with. Because our validators may need to reference values in other fields, we extend a simple interface called ValidatingField, which has getName() and getValue() methods to retrieve the name of the field and its value, respectively. This is useful for fields in a change password dialog, for example, where a confirmation field is expected to match the initial password field. The MatchingFieldValidator can do this easily, so long as the other field implements the ValidatingField interface.

JValidatingField exposes the same constructors as any JTextField but adds a name argument, which is used to describe the field in text messages presented to the user. This is a requirement of the ValidatingField interface we implement. The addValidator() and removeValidator() methods delegate their functionality to the SwingFieldValidator assigned to the field. Notice that, aside from the name argument, the constructors reflect the same options available in any JTextField, so replacing an existing JTextField with a JValidatingField is a simple matter of adding a name as the first argument in any constructor.

As always, the amount of space we have in print is limited, so I'll give you a whirlwind tour of each of the validator implementations I've provided. You can download them all from www.java-pro.com and use them as you see fit. We'll work through them in the order you see them in the class diagram.

Continued from page 54.

```
  {
    this(name, new PlainDocument(), text, 0);
  }

  public JValidatingField(String name, String text, int col)
  {
    this(name, new PlainDocument(), text, col);
  }

  public JValidatingField(String name, Document doc, String
    text, int col)
  {
    super(doc, text, col);
    this.name = name;
    setInputVerifier(verifier =
      new SwingFieldVerifier(this, name));
  }

  public String getName()
  {
    return name;
  }

  public String getValue()
  {
    return getText();
  }

  public void addValidator(FieldValidator validator)
  {
    verifier.addValidator(validator);
  }

  public void removeValidator(FieldValidator validator)
  {
    verifier.removeValidator(validator);
  }
}
```

Listing 3 Learning the Alphabet The AlphaFieldValidator class expects to see valid alphabetical values.

```
import java.awt.*;
import javax.swing.*;

public class AlphaFieldValidator
  extends AbstractFieldValidator
{
  public AlphaFieldValidator()
  {
    this("This field must be an alphabetical value");
  }

  public AlphaFieldValidator(String message)
  {
    super(message);
  }

  public boolean isValid(String value)
  {
    if (isNull(value)) return false;
    for (int i = 0; i < value.length(); i++)
    {
      if (!Character.isLetter(value.charAt(i)))
        return false;
    }
    return true;
  }
}
```

Listing 4 A Number of Values The AlphaNumFieldValidator class expects to see alphanumeric values.

```
import java.awt.*;
import javax.swing.*;

public class AlphaNumFieldValidator
  extends AbstractFieldValidator
{
  public AlphaNumFieldValidator()
  {
    this("This field must be an alphanumeric value");
  }

  public AlphaNumFieldValidator(String message)
  {
    super(message);
  }

  public boolean isValid(String value)
  {
    if (isNull(value)) return false;
    for (int i = 0; i < value.length(); i++)
    {
      if (!Character.isLetterOrDigit(value.charAt(i)))
        return false;
    }
    return true;
  }
}
```

Become a charter subscriber NOW!

PROGRAM BETTER AND FASTER. Subscribe Now.

$19.95 12 issues annually

Avoid 5 Common Release Bugs
Visual C++
Developers Journal
8 Design Tips for Building COM Interfaces
Delve Into COM+ Events
Throw and Catch COM Exceptions
Manipulate 3-D Objects

Name

Address

City State Zip

E-mail

O Payment Enclosed O Bill Me O Charge My: O Mastercard O VISA O AMEX

Card No. Exp. Date

Signature

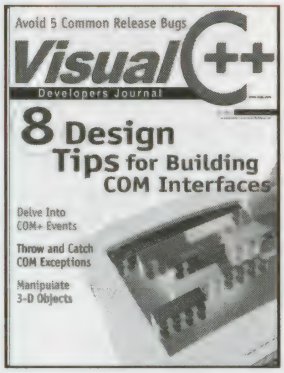
One year is 12 monthly issues. Please allow 4–6 weeks for delivery of first issue. International subscriptions must be payable in U.S. dollars plus postage. Canada and Mexico add $18/year, all other countries add $44/year for airmail delivery. Canadian GST included.

4927

Become a charter subscriber NOW!

PROGRAM BETTER AND FASTER. Subscribe Now.

$19.95 12 issues annually


Avoid 5 Common Release Bugs
Visual C++
Developers Journal
8 Design Tips for Building COM Interfaces
Delve Into COM+ Events
Throw and Catch COM Exceptions
Manipulate 3-D Objects

Name

Address

City State Zip

E-mail

O Payment Enclosed O Bill Me O Charge My: O Mastercard O VISA O AMEX

Card No. Exp. Date

Signature

One year is 12 monthly issues. Please allow 4–6 weeks for delivery of first issue. International subscriptions must be payable in U.S. dollars plus postage. Canada and Mexico add $18/year, all other countries add $44/year for airmail delivery. Canadian GST included.

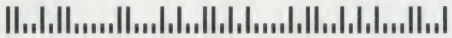
4927

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Visual C++
Developers Journal
P.O. Box 59289
Boulder, CO 80323-9289

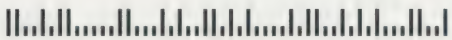


NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Visual C++
Developers Journal
P.O. Box 59289
Boulder, CO 80323-9289



Code Faster, CODE Smarter

Your Technical Guide to Faster, Better VC++ Code

You're an experienced, savvy Visual C++ professional, and *VCDJ* is your indispensable tool for faster, smarter code!

There's no better independent guide to hardcore, professional Visual C++ programming. You can't afford to miss one issue, or even one article, of *VCDJ's* practical advice for corporate development. You'll find timesaving tips and proven techniques for building server-side components, client executables, Web apps, or anything Visual C++ can create. Each issue will bring you hands-on, code-intensive articles that help you:

- Turbocharge your Visual C++ apps' performance
- Choose the right features and functions to balance speed and functionality
- Expose the right COM interfaces in the right way
- Master the latest Visual C++ techniques and tools
- Control memory allocation like a pro
- Take advantage of new programming paradigms
- Access data fast
- Harness the power of XML

Visual C++ Developers Journal

Don't miss a single issue.

Order online

www.vcdj.com

Order by phone: 1-800-848-5523

Outside the U.S. : 1-650-833-7100

VCDJ is another high-quality, how-to publication from Fawcette Technical Publications

VISUAL C++ IS A REGISTERED TRADEMARK OF MICROSOFT CORP. ALL OTHER PRODUCT NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS

The AlphaFieldValidator (Listing 3) and AlphaNumFieldValidator (Listing 4) classes expect to see valid alphabetical and alphanumeric values, respectively. The BlankFieldValidator (Listing 5) is typically the first one used and it watches for fields that have no value. Since the order in which you add validators is relevant to the order in which they are tested, you'll want to keep this in mind when you add them to JValidatingField. The DecimalFieldValidator (Listing 6) accepts any decimal value that qualifies as a Double. It does its validating by calling the parseDouble() method and catching the NumberFormatException.

Listing 5 Searching for Nothing The BlankFieldValidator watches for fields that have no value.

```
public class BlankFieldValidator
  extends AbstractFieldValidator
{
  public BlankFieldValidator()
  {
    super("You must provide a value for this field");
  }

  public BlankFieldValidator(String message)
  {
    super(message);
  }

  public boolean isValid(String value)
  {
    if (isNull(value)) return false;
    return value.length() > 0;
  }
}
```

Listing 6 A Decimal in the Right Place The DecimalFieldValidator accepts any Double decimal value by using the parseDouble() method and catching the NumberFormatException.

```
public class DecimalFieldValidator
  extends AbstractFieldValidator
{
  public DecimalFieldValidator()
  {
    this("This field must be a Decimal value");
  }

  public DecimalFieldValidator(String message)
  {
    super(message);
  }

  public boolean isValid(String value)
  {
    if (isNull(value)) return false;
    try
    {
      Double.parseDouble(value);
      return true;
    }
    catch (NumberFormatException e)
    {
      return false;
    }
  }
}
```

Listing 7 Checking the Format The FormatFieldValidator class ensures that the field value matches the given Java Format implementation.

```
import java.text.*;

public class FormatFieldValidator
  extends AbstractFieldValidator
{
  protected Format format;

  public FormatFieldValidator(Format format)
  {
    this(format, "You must provide valid text format");
  }

  public FormatFieldValidator(Format format,
    String message)
  {
    super(message);
    this.format = format;
  }

  public boolean isValid(String value)
  {
    if (isNull(value)) return false;
    try
    {
      format.parseObject(value);
      return true;
    }
    catch (ParseException e)
    {
      return false;
    }
  }
}
```

Listing 8 A Similar Class The IntegerFieldValidator class performs pretty much the same function as the DecimalFieldValidator class, except IntegerFieldValidator uses the Integer.parseInt() method instead of the parseDouble() method.

```
public class IntegerFieldValidator
  extends AbstractFieldValidator
{
  public IntegerFieldValidator()
  {
    this("You must enter an Integer value");
  }

  public IntegerFieldValidator(String message)
  {
    super(message);
  }

  public boolean isValid(String value)
  {
    if (isNull(value)) return false;
    try
    {
      Integer.parseInt(value);
      return true;
    }
    catch (NumberFormatException e)
    {
      return false;
    }
  }
}
```

GO FROM E-CONCEPT TO E-BUSINESS IN NO TIME FLAT.

Featuring

VisualCafé | StructureBuilder | TopLink | Macromedia Dreamweaver

We've acquired and licensed some of the best technologies on the planet
and created an integrated suite that can cut weeks,
maybe even months, off your application development schedule.

Introducing WebGain Studio.™

It's a complete, end-to-end, browser-to-database Java development solution.
And from design to code to deployment, there's nothing more powerful or productive.

Check it out today at **www.webgain.com**.

webGAIN™

©2000 WebGain, Inc. WebGain, WebGain Studio, VisualCafé, StructureBuilder and TopLink are registered trademarks of WebGain, Inc.
Dreamweaver is a registered trademark of Macromedia, Inc.

Listing 9 It's a Match The MatchingFieldValidator looks for a reference to another field that implements the ValidatingField interface and verifies that the value matches.

```
import java.awt.*;
import javax.swing.*;

public class MatchingFieldValidator
  extends AbstractFieldValidator
{
  protected ValidatingField field;

  public MatchingFieldValidator(ValidatingField field)
  {
    this(field, "This field must match the field named
      " + field.getName());
  }

  public MatchingFieldValidator(ValidatingField field,
    String message)
  {
    super(message);
    this.field = field;
  }

  public boolean isValid(String value)
  {
    if (isNull(value)) return false;
    return value.equals(field.getValue());
  }
}
```

Listing 10 On the Range The RangeFieldValidator class looks for an integer value that falls within a specified range.

```
import java.awt.*;
import javax.swing.*;

public class RangeFieldValidator
  extends AbstractFieldValidator
{
  protected int min = Integer.MIN_VALUE;
  protected int max = Integer.MAX_VALUE;

  public RangeFieldValidator(int min, int max)
  {
    this(min, max, "You must provide a value between
      " + min + " and " + max);
  }

  public RangeFieldValidator(int min, int max,
    String message)
  {
    super(message);
    this.min = min;
    this.max = max;
  }

  public boolean isValid(String value)
  {
    if (isNull(value)) return false;
    try
    {
      int val = Integer.parseInt(value);
      return val >= min && val <= max;
    }
    catch (NumberFormatException e)
    {
      return false;
    }
  }
}
```

FormatFieldValidator (Listing 7) checks to see that the field value matches a given Java Format implementation by calling its parseObject() method and catching a ParseException. The constructor lets you pass in any class that implements the Format interface, such as the standard DateFormat, MessageFormat, or NumberFormat.

The IntegerFieldValidator (Listing 8) does the same thing as the DecimalFieldValidator, but uses the Integer.parseInt() method, catching the NumberFormatException. The MatchingFieldValidator (Listing 9) expects a reference to another field that implements the ValidatingField interface and verifies that the value matches that of the other field. You can use this with password confirmation fields. The RangeFieldValidator (Listing 10) expects an integer value that falls within a specified range of values, which you can specify in the constructor.

By providing a uniform mechanism that lets you add or remove validators in any combination, JValidatingField can reduce your field validation workload considerably

Building a useful inventory of reusable FieldValidator instances is easy enough. SwingFieldVerifier reduces the number of classes you need to implement, and JValidatingField takes care of your interface needs more effectively, making it easier to deliver robust user interfaces to your end users. By providing a uniform mechanism that lets you add or remove validators in any combination, JValidatingField can reduce your field validation workload considerably. As always, have fun with it. ■

Claude Duguay is vice president of engineering at 360 Powered Corp. He has developed software for more than 20 years. He is passionate about Java, reads about it incessantly, and writes about it as often as possible. You can reach him at claude@360.com.

Go Online

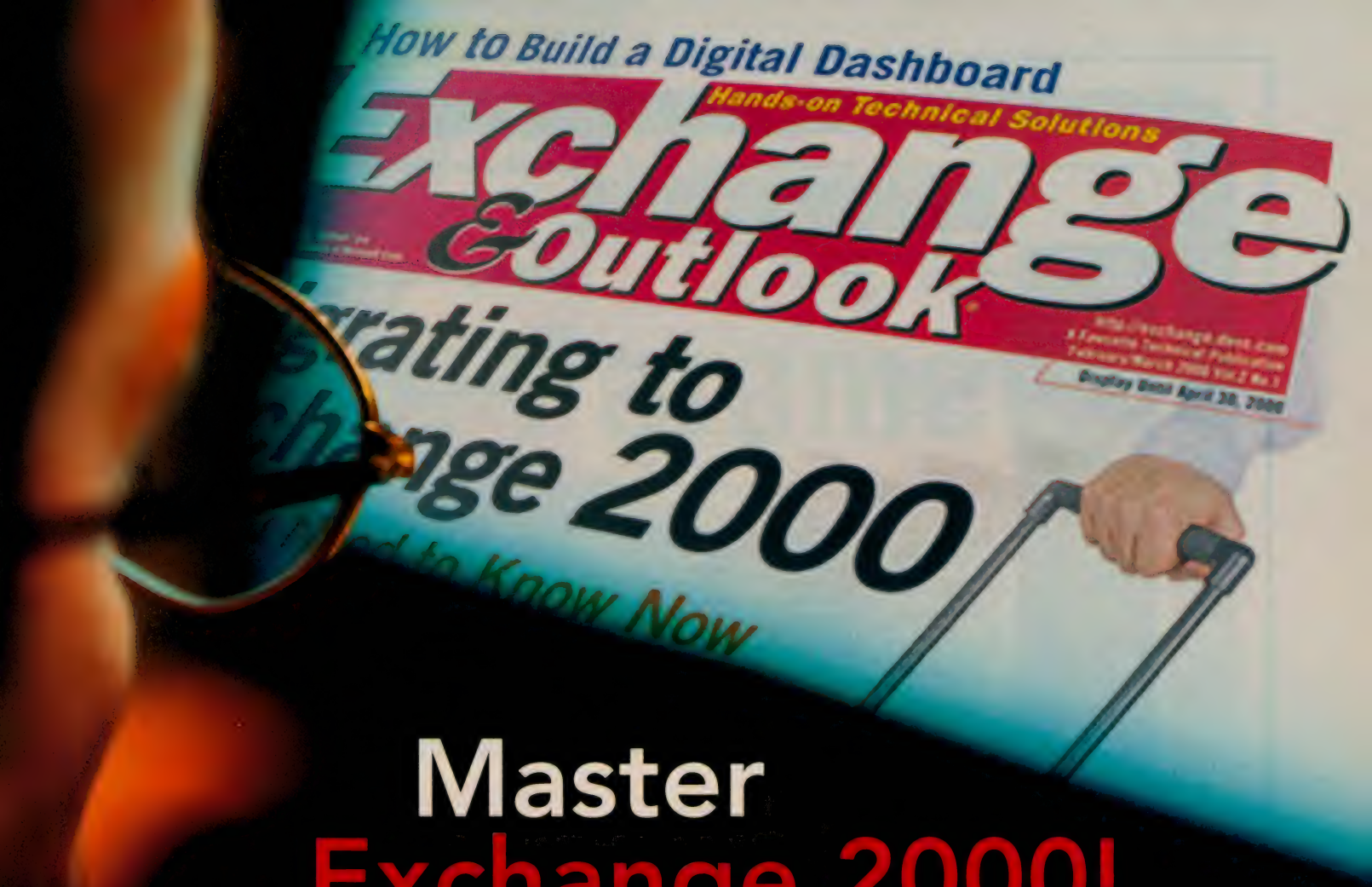
Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0103 Download all of the code for this issue of *Java Pro*.

JP0103VC Download the code for this article.

JP0103VC_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.



Master Exchange 2000!

With the First and Foremost Exchange Authority

The ONLY independent Exchange and Outlook magazine, written by the experts exclusively for IT professionals, administrators, managers and corporate developers using the latest Microsoft technologies.

Packed with hands-on, in-depth technical information, each page of *Exchange & Outlook* contains proven strategies and no-nonsense tips to help you successfully use these technologies for your enterprise-wide challenges:

- Successful deployment of Exchange 2000
- Critical, independent evaluations of new products
- Real-world administration solutions to real-world problems
- Code you can actually use
- Proven deployment strategies for custom and collaborative apps
- Groupware projects and strategies, and more

Save 52% off the cover price!

Subscribe online

http://exchange.devx.com/in.asp

Call today 1-800-848-5523 and pay only $39.95 for a full year: twelve issues in all.

One year is based on twelve issues. Annual cover price is $83.40. Please allow four-to-six weeks for delivery of first issue. International subscriptions must be paid in U.S. dollars plus postage. Canada/Mexico, add $18/year; other countries add $44/year for airmail. Canadian GST included.

Exchange & Outlook magazine is another high-quality, how-to technical publication published by Fawcette Technical Publication. Exchange and Outlook are trademarks of Microsoft Corp. All other product names herein are the properties of their respective owners.

Upgrade NOW

Premier Club Delivers:

- Savings on software
- Access to the DevX development article and code archive
- Live member-only events

Join the Premier Club for $3.95 per month and your first 30 days are free!

From the DevX home page click on JOIN

Need Development Know-How?

- Tap into 1000's of programming tips
- Ask the experts
- Browse tons of content and code from top IT publications
- Join live developer forums
- Access over 2,000 product reviews
- Purchase and download leading development tools
- Join Premier Club for exclusive benefits

Your search stops at DevX.

Get the unbiased information and tools you need NOW from the leading information resource for IT professionals developing business applications. We connect developers with objective, technical information and tools to get the job done—code samples, technical articles, discussions, learning services, and other resources aimed at helping you solve your toughest development challenges.

To learn more, visit:

www.devx.com

DevX is a trademark of DevX.com, Inc.

devx development exchange

Java To Go

Pesky Parameters

If the Real-Time Specification for Java *definitions sound like theory only, check out how its parameter classes describe real-time threads*

BY RICK GREHAN

Last time I explored the *Real-Time Specification for Java* (*RTSJ*) and concentrated on scheduling. I examined rate monotonic analysis, the default schedule analysis algorithm in the *RTSJ* scheduler (see "Right on Schedule," *Java Pro,* February 2001). I also introduced the *RTSJ* notion of a "schedulable object"—runnable entities under control of the scheduler that could be either real-time threads or asynchronous event handlers.

I mentioned that the *RTSJ* defined a set of parameter classes whose instantiated objects could be associated with "schedulable objects" for the purpose of passing to the scheduler important characteristics of those schedulable objects. Now I'm going to take a close look at those parameter classes, because their makeup can tell us a lot about the aspects of real-time threads, which they are meant to describe.

There are four of these parameter classes, and it's important to recognize that they are "data" classes. That is, they are not accompanied by any methods beyond the getxx() and setxx() methods used to manipulate their members. As such, they are more like C/C++ structs than anything else.

Instances of these objects are bound to a schedulable object through the schedulable object's constructor. When you register a schedulable object with the scheduler, the scheduler can examine its associated parameter objects for the purpose of scheduling decisions.

You can bind a single instance of a parameter class to multiple schedulable objects. So, you could create a single instance of, say, PeriodicParameters and pass it to several schedulable object constructors; they would all share that single instance. The relationship between parameter classes and schedulable objects can therefore be one to many.

ReleaseParameters

ReleaseParameters is an abstract class with two direct, concrete descendants: APeriodicParameters and PeriodicParameters. The base class defines four parameters. They are:

- cost: This parameter indicates the cost of the schedulable object in terms of CPU time. In other words, this is the maximum amount of time a schedulable object receives per period (or per interval). Strictly speaking, cost is a reference to an object of class RelativeTime and expresses time units per interval (see sidebar "Give Me Some Time"). This example might make the terminology clear. Suppose a real-time thread must execute once every second. Each time it executes, the thread takes 0.5 seconds to do whatever its job is. In this case, the schedulable object's period is 1 second, and its cost is 0.5 seconds. Such a real-time thread is called a "periodic" thread, and you'll see in a moment that another parameter class provides the means by which the scheduler can determine the thread's period.
- deadline: Also a reference to a RelativeTime object, this parameter indicates the latest permissible deadline for the task to complete. (The time given by deadline is an interval, rather than an absolute point in time. So, the "real" deadline is found by adding the interval expressed in deadline to the starting time of the schedulable object's period.)
- overrunHandler: This is a reference to another schedulable object. In particular, overrunHandler references an AsyncEventHandler, which is invoked if the schedulable object governed by this particular instance of ReleaseParameters overruns its allotted time (that is, exceeds its cost).
- missHandler: This field is also a reference to an AsyncEventHandler. The AsyncEventHandler is invoked if the schedulable object governed by this instance of ReleaseParameters is still executing after the time given by deadline passes. (So, if a real-time thread's run() method is still executing after the time given by deadline passes, the missHandler AsyncEventHandler is called.)

The PeriodicParameters and APeriodicParameters subclasses of ReleaseParameters define the characteristics of two sorts of schedulable objects. Schedulable objects with which an object of class PeriodicParameters is associated—you guessed it—execute periodically. (As such, PeriodicParameters will almost always be associated with real-time threads.) For example, a real-time thread that executes once per second to read a temperature and update an onscreen thermometer would be a periodic real-time thread.

Give Me Some Time

Many of the parameters I've mentioned are objects of type RelativeTime, which is one of several classes that encapsulate time (and time intervals) in the *RTSJ*. All of the so-called "time" classes are derived from the abstract class HighResolutionTime. This class—and its subclasses—can express time down to nanosecond resolution. (That's the resolution that the classes can represent; the resolution of time that can be delivered by system hardware is, of course, system dependent.)

The classes provide methods to get and set the millisecond and nanosecond components, for adding and subtracting high-resolution times, for testing equality, for converting objects from one class to another, and so on. The three classes derived from HighResolutionTime are:

AbsoluteTime: The name of this class says it all. An AbsoluteTime object represents a measurement of time as calculated from midnight, January 1, 1970 GMT. (This representation was chosen to be compatible with the time represented in the java.util.Date class.) So, for example, this code:

```
now = new
AbsoluteTime(
   System.currentTimeMillis(), 0);
System.out.println(now);
```

will yield output similar to:

```
Mon Dec 11 11:15:05 EST 2000+332ms
```

Notice the postfix expression for the millisecond accuracy. If you were to change the preceding code slightly:

```
now = new AbsoluteTime(
   System.currentTimeMillis(), 100);
System.out.println(now);
```

the output would look like:

```
Mon Dec 11 11:15:05 EST 2000+332000500ns
```

thereby "capturing" the increased nanosecond accuracy.

RelativeTime: Objects of this class encapsulate a point in time as measured from some other point (typically, it represents a time as measured from the present instant). In short, this class is used to express a duration of time, rather than a fixed point in time. For example, a periodic real-time thread's cost—which is a measurement of how much time the thread will execute within a given interval—is expressed in RelativeTime.

RationalTime: Objects of the RationalTime class describe events that occur with a certain frequency within a given time interval. You can think of a RationalTime object as consisting of three numbers, represented by *f*, *m*, and *n*. The *m* and *n* numbers are the millisecond and nanosecond components of a high-resolution time; they define the time interval. The *f* number is the frequency; the number of times the event will occur. So, a RationalTime object is an ideal mechanism for encapsulating the behavior of a periodic real-time thread (among other things).

The PeriodicParameters subclass adds two parameters to the base ReleaseParameters class. They are:

- start: This is the starting time of the first period for this schedulable object. For example, if this parameters class governs a real-time thread, this field indicates when the thread's first invocation is possible. This field is an object of type RelativeTime, hence the starting time of the first period is the current time plus the RelativeTime given by start.
- period: This indicates the interval of the schedulable object. Technically, it is the time between "successive unblocks" of the thread's waitForNextPeriod() method. (The waitForNextPeriod() method is defined within the RealTimeThread class and is used by real-time threads that have an associated reference to a PeriodicParameters object. When a real-time thread calls the waitForNextPeriod() method, that method will block until the start of the next period.)

Meanwhile, schedulable objects that are aperiodic do not execute on regular time intervals. The APeriodicParameters class describes the characteristics of aperiodic schedulable objects. APeriodicParameters redefines the base ReleaseParameters' cost field. Since there is no notion of a "period" or "interval" in an aperiodic schedulable object, the cost field is redefined simply to mean the time it takes the schedulable object to execute.

Finally, there is a peculiar sort of schedulable object that we will call a sporadic schedulable object (in keeping with *RTSJ* terminology). A sporadic schedulable object is aperiodic—it isn't scheduled at a fixed interval. However, there is a minimum duration of time between each invocation of a sporadic schedulable object. In other words, you can say of a sporadic schedulable object: "I know that after an invocation of this object, the next invocation will occur no sooner than X seconds" (where X can be fractional).

Hence, the *RTSJ* provides the SporadicParameters class, which extends the APeriodicParameters class with a single parameter:

- minInterarrival: Refers to a RelativeTime object that specifies the lower bound on the time between consecutive invocations of the schedulable object.

SchedulingParameters

SchedulingParameters is an abstract class whose concrete subclasses carry information that plays an important role in the scheduling of the schedulable object. The *RTSJ* defines a subclass and a "grandchild" subclass to SchedulingParameters. The subclass is PriorityParameters, and its subclass is ImportanceParameters.

The structure of PriorityParameters is simple. Instances of this class carry a single member: an integer value that represents the thread's priority.

The ImportanceParameters class is provided for those circumstances in which the scheduler must select among a set of schedulable objects of equivalent priority. Therefore, ImportanceParameters extends PriorityParameters by adding another integer value, importance. This allows the application to assign to each schedulable object a number that indicates the significance of that object, relative to other schedulable objects of the same priority. In a sense, this adds another level to priorities—sort of a priority within a priority. When faced with scheduling two real-time threads of the same priority, the scheduler will schedule the task with the higher importance.

ProcessingGroupParameters

The ProcessingGroupParameters class allows a developer to associate a set of parameters to a group of schedulable objects (usually aperiodic and sporatic real-time threads). The set of parameters

devX marketplace

development exchange

Your #1 Source for Development Tools!

Software

Order and Download Your Favorite Tools Now!

APEX True DBSuite 6.0
The complete solution for database front-end applications
Format: Download
$679
Upgrade for $559

Desaware SpyWorks
Extend the capabilities of VB to rival those of C++
Standard Edition
Format: Download
$169

Sheridan™ Components Suite™
Combines the best of Sheridan's ActiveX components in one package
Format: Download
$319

eHelp RoboHelp Office 9.0
Provides a powerful native environment for authoring new content—works hand-in-hand with printed documentation
Format: Download
$899

LGB Systems XControls
Quickly develop scalable and robust database applications
Enterprise Edition
Format: Download
$329

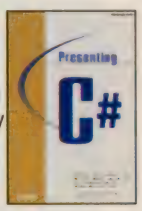
VideoSoft® VSFlexGrid Pro® 7
Gives VB developers a comprehensive feature set for display and presentation of dynamic data
Format: Download
$249

Books

Get up to speed with the best books to help you do your job better and faster.

Enterprise Application Integration with XML and Java
Includes techniques for building flexible, extensible EIA solutions
By JP Morgenthal
$35.99

Presenting C#
Get up to speed and begin creating next-generation applications and Web services today
By Christoph Wille
$20



Advanced Visual Basic 6 : Power Techniques for Everyday Programs
Learn how to write better VB code with new, advanced techniques.
By Matthew J. Curland
$31.96

A Preview of Active Server Pages+
Includes a supporting Web site with features, changes and updates to ASP+.
By Alex Homer, David Sussman, Richard Anderson & Robert Howard
$27.99

Check out our complete book inventory covering:

.NET	Databases	SQL Server
Access	Dynamic HTML	Visual Basic
Active Server Pages	Enterprise Development	Visual InterDev
ActiveX	Exchange and Outlook	Web Development
ADO	Java	Windows 2000
C++	Linux	XML
Certification	MTS	
COM/DCOM	Security	

New Releases!

GO

Also check online for great deals on these other Microsoft® products:

- Microsoft® Commerce Server 2000
- Microsoft® Host Integration Server 2000
- Microsoft® Internet Security and Acceleration Server 2000
- Microsoft® Application Center 2000
- Microsoft® SQL Server 2000 Standard Edition
- Microsoft® SQL Server 2000 Enterprise Edition
- Microsoft® BizTalk Server 2000
- Microsoft® Access 2000
- Microsoft® Outlook 2000
- Microsoft® Project 2000

Microsoft®

To Order:

Go online at http://marketplace.devx.com or, e-mail us at marketplace@devx.com

Please call us directly for personalized service on higher volume pricing and purchases, or for additional information.

OVERNIGHT EXPRESS DELIVERY

1- 800-848-5523
(outside the US call 1-650-833-7100)
(8:30am - 5:30pm M-F, Pacific Time

Order Online and Save!

http://marketplace.devx.com

Or call 800.848.5523 or 650.833.7100 (outside the USA)

DevX.com MarketPlace prices are subject to change without notice. The Development Exchange, DevX, and DevX.com are trademarks of DevX.com, Inc. All products listed are registered marks or trademarks of their respective owners.

enforces a guarantee (by the scheduler) on the group. That guarantee is that the scheduler will not give more time per period than is indicated by the cost to objects in the group.

The parameters in a ProcessingGroupParameters object are an amalgam of those in the ReleaseParameters class and its subclass, PeriodicParameters. In short, the parameters in the ProcessingGroupParameters class are start, cost, deadline, overrunHandler, and missHandler, and their meanings are as I've already described them.

With a moderately reasonable x86 system and access to the Internet, you can download everything you need to take a dive into a beta real-time Java implementation

MemoryParameters

The MemoryParameters class is somewhat unique in the group of parameter classes because it does not work with specific scheduling characteristics (such as cost and priority). The MemoryParameters class allows a schedulable object to pass information about its memory utilization to the scheduler and garbage collector. (MemoryParameters objects may only be associated with RealTimeThread and AsyncEventHandler objects.) The parameters in a MemoryParameters object are:

- maxMemoryArea: This specifies the limit on the amount of memory the schedulable object will allocate in its associated memory area. The *RTSJ* allows you to associate a MemoryArea object with a real-time thread object. Briefly, a MemoryArea object can describe a region of memory storage with specific characteristics and restrictions on the objects stored in the area. This field sets the upper limit on the amount of memory the object will allocate in its associated memory area. (The default memory area assigned to a real-time thread is the heap.)
- maxImmortal: This specifies the limit on the amount of memory the schedulable object can allocate in the immortal memory area. (Objects created in the immortal memory area persist for the lifetime of the application—they never die.)
- allocationRate: This value specifies the bytes-per-second allocation rate in the heap allowed for the associated schedulable object. The garbage collector can use this information to pace itself.

The Real Thing

If all this sounds like theory, and that only those who happen to have a real-time operating system (RTOS) on hand can begin experimenting with real-time Java, cheer up. If you have a moderately reasonable x86 system on hand (sorry, Mac users) and access to the Internet, you can download everything you need to take a dive into a beta real-time Java implementation. (I hope that by the time you read this, the implementation will have progressed beyond the early beta stage it is in as I write this.)

Your first stop is the QNX Web site: www.qnx.com. QNX has made available a download of its QNX RTP (real-time platform). This is a "self-hosted" development platform that runs on a variety of x86 processors. (It also supports several target processors; see the Web site for system specifications. Certainly, if your system is running Windows 98, Me, NT, or 2000, it will probably work.) You can even install QNX RTP so as to coexist with your existing Windows 9x/Me or NT/2000 operating system. The only caveat is that the QNX installation requires a FAT filesystem. NTFS won't work. Also, the version available at the time of this writing was incompatible with SCSI hard disks.

Running the installation is about as simple as it gets. I won't bore you with the details of it here. Although the installation instructions recommend a 600 MB "workspace" (the region in the FAT filesystem where QNX resides), I created a 400 MB workspace. For what I was doing it seemed more than enough (I did not install all the QNX tools that are available). What you end up with is a sort of "dual-boot" system that you can boot either under Windows or QNX.

Next, go to IBM's VisualAge Micro Edition Web site: www.embedded.oti.com. There you'll find—among other things—the downloadable files for the real-time extensions beta code of IBM's J9 virtual machine. Download the package (it includes the library and J9 executable for the QNX RTP) and follow the instructions to move the files into your QNX filesystem. Even if you don't set up Internet support on QNX, you can still download the J9 .ZIP file on the Windows "side," then reboot under QNX and simply copy the file from the Windows partition into the QNX partition to unzip it. This is easily done because the Windows partition maps to the /fs/hd0-dos subdirectory in QNX. Unzip the file using the QNX unzip utility. (You can use Winzip on the Windows side, but I found that Winzip did not preserve the case of the letters in the filenames, something that Windows doesn't concern itself with, but QNX does. Also, do *not* try to run the J9 virtual machine from the Windows partition. When I tried this, I found that I could not set the file's execute attribute.)

Several example files are available with the J9 download. You can either install Java on QNX and compile the sourcefiles in the QNX environment, or you can compile the files on the Windows side and move the resulting classfiles for execution in the QNX partition. In the latter case, you can even download and install IBM's free VisualAge Micro Edition Individual Package and use that as the compiler. Either way, there is now little excuse for not experimenting with the *RTSJ*. ■

Rick Grehan is a technical evangelist with Compuware/NuMega. Rick was a senior editor at *BYTE* magazine (where he wrote the JavaTalk column) and at *Computer Design* magazine. He can be reached at rgrehan@top.monad.net.

Go Online

Use the following DevX Locator+ code at www.java-pro.com to go directly to this related resource.

JP0103JG_T Read this feature online. DevX Premier Club membership is required.
Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

devX eLearning

Real World Learning Solutions for IT Professionals *Led by the Industry Experts*

DevX is the leading site IT developers turn to for the latest, most comprehensive information available on software development for corporate applications. Our eLearning offerings enhance our commitment to IT professionals around the world by providing both live and Web-based eLearning events to nearly any desktop. DevX eLearning offers:

Online, Live Events

Upcoming events, powered by InterWise, include:

- Microsoft Exchange 2000 Series
- Visual Basic® Tool & Techniques Series

Web-based IT Training

We offer a broad range of online courses in technical areas such as Flash, Java, Visual Basic and more!

Customized Corporate eLearning Portals

DevX can optimize your company's learning resources by providing training tailored specifically to your organization's needs.

The classroom meets new media and gives you a new, convenient, and customized way to learn.

Register Today for These Special Online Events

Hurry—space is limited!

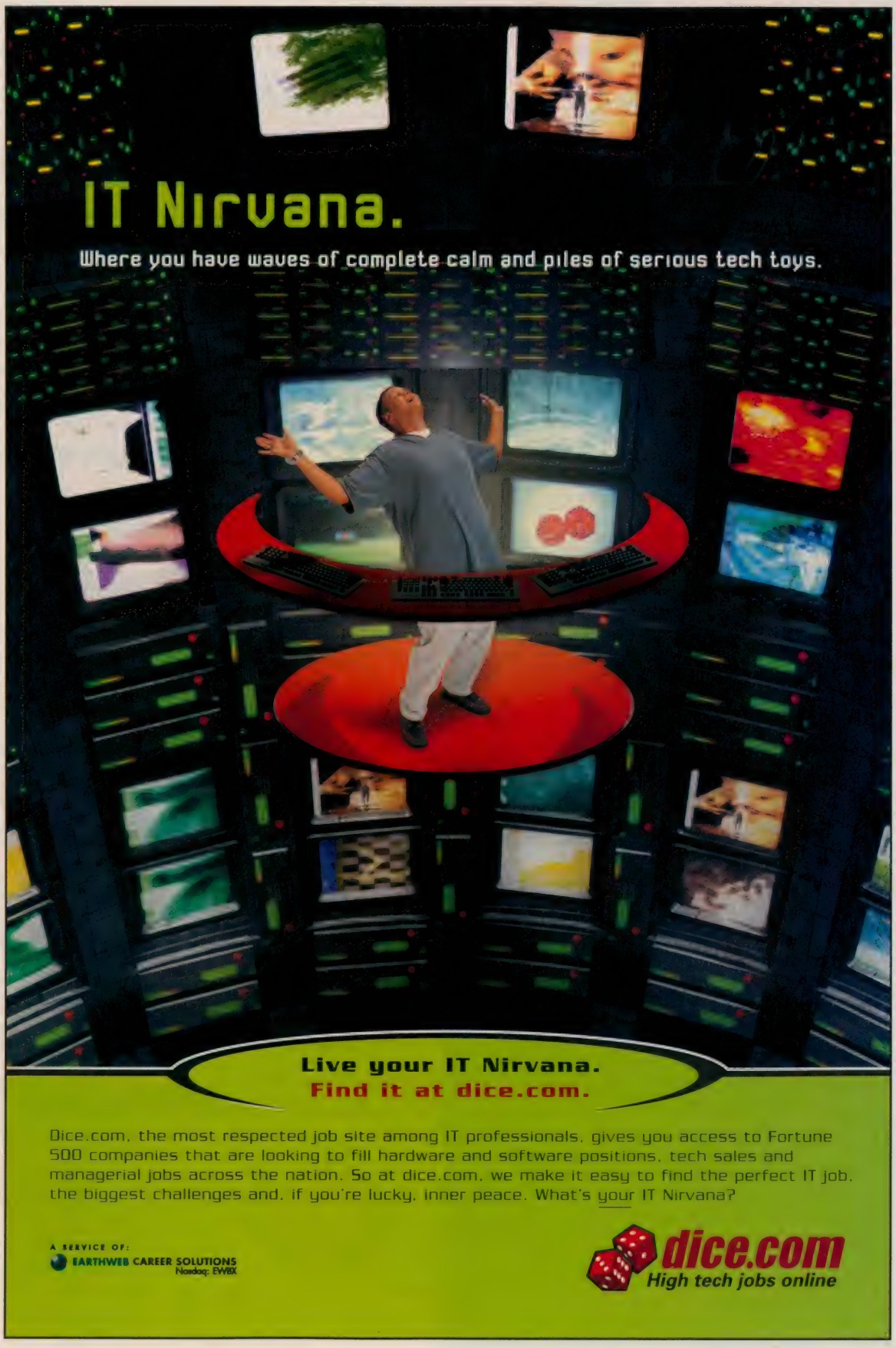
www.devx.com/elearning

- Call **1-888-963-DEVX**
  (Outside the US call 650-566-2000)
- Or email us at
  eLearning@devx.com

devx

development exchange

The Development Exchange, DevX, and DevX.com are registered trademarks of DevX.com, Inc. All other product names listed herein are trademarks or registered trademarks of their respective owners.


IT Nirvana.
Where you have waves of complete calm and piles of serious tech toys.
Live your IT Nirvana.
Find it at dice.com.
Dice.com, the most respected job site among IT professionals, gives you access to Fortune 500 companies that are looking to fill hardware and software positions, tech sales and managerial jobs across the nation. So at dice.com, we make it easy to find the perfect IT job, the biggest challenges and, if you're lucky, inner peace. What's your IT Nirvana?
A SERVICE OF:
EARTHWEB CAREER SOLUTIONS
Nasdaq: EWBX
dice.com
High tech jobs online

CareerLink
For IT Development Professionals

FIND A COOL JOB

(Actually, over 100,000...)

Real jobs.
Cool jobs.
In all fields.
All in one place online.

SEARCH JOB
POST A RESUME
JOB MATCH AGENT
EMPLOYER PROFILE
ARTICLES
MESSAGE BOARD

careermag.com
Cool Jobs. Hot Content.

GET YOUR CAREER CLICKING!

CareerLink
For IT Development Professionals

Recruit Professional Programmers
Through the Leading IT Development Magazines

To place an ad in this section, please contact:

Kathy Bazan
Western Regional Manager
Phone: (650) 833-7148
E-mail: kbazan@fawcette.com

Jean Anne Mahler
Eastern Regional Manager
Phone: (650) 833-7109
E-mail: jamahler@fawcette.com

Bored?

Wouldn't you rather be developing KILLER applications?

Why develop boring database applications for a corporate giant when you could be developing killer wireless, messaging, speech and XML applications for a Chicago-based startup? Centerpost is using XML to develop the next generation of electronic messaging services and we're looking for more people to join our top-notch product teams.

If you have experience with VB, ASP, VC++, Java, Windows 2000, and XML, you need to give us a call. We're seeking the best and the brightest to join our product development, product management and product operations teams. We offer:

- Big, open office in downtown Chicago
- Casual work environment
- Soda, juice, treats of all kinds
- Foosball, PS2® and a ton of Nerf® toys
- Generous salary/options package & 3 weeks paid personal time
- Full medical, dental, & vision benefits
- Transit benefit
- Voluntary life insurance
- Long & short-term disability insurance

We're backed by Motorola, Draper Fisher Jurvetson and Portage Venture Partners and we're already working with several major corporations. Get in on the ground floor of our explosive company! To apply, please email your resume to jobs@mail.centerpost.com or, for more information, visit our website.

[•] centerpost
think outside the inbox℠

200 West Monroe Street, Suite 1400
Chicago, IL 60606-5015
www.centerpost.com

CareerLink
For IT Development Professionals

Build

Better, Smarter Web Sites

All the Tools, Tips and Techniques You Need to Create Your Web Site Quickly and Easily!

Save 42% off the newsstand price!

For just $29.95 per year, you'll get a new CD every quarter, packed with:

- How-to articles from Visual Basic Programmer's Journal, WEB Builder, Java™ Pro, Technical Guide to Visual Programming, Enterprise Development, Getting Starte with Visual Basic, and more.
- New tools and utilities, complete with the latest versions.
- Product reviews, software and links
- Plus dozens of tips, basics, code... and more!

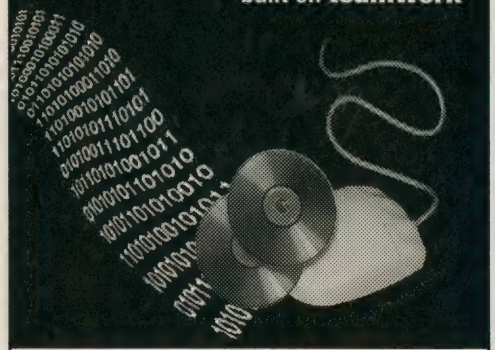
To Subscribe to Web Builder CD:
E-mail us at webcs@magserv.com
Visit us at marketplace.devx.com/moretools.asp
Call us at 1-800-848-5523
Or write to us at P.O. Box 15025
N. Hollywood, CA 91615

FTP FAWCETTE TECHNICAL PUBLICATIONS

WEB Builder is a product of Fawcette Technical Publications, Inc.
All product names herein are the property of their respective owners.

GROWTH

e-commerce solutions, built on **teamwork**



Growth, Performance and Strength converge in Grand Rapids, Michigan where a diverse, dynamic and technologically advanced economy combines with affordable living, friendly communities and the lifestyle options you desire. Here creative ideas, career challenge and advanced solutions are built on our $2 billion plus leadership in food service distribution.

At GFS, we fulfill our mission of providing the highest quality products and services by implementing dynamic and innovative software, middleware and hardware technologies. Now, we're looking for IS professionals with the following skills to join us:

- **J2EE**
- **JAVA**
- **JMS MQSeries**
- **WebLogic**
- **XML, HTML**
- **Manugistics**
- **Constellar**
- **PMI**
- **Business Objects**
- **VPN**
- **Oracle**
- **UNIX Sun Solaris**
- **C/C++**
- **VB**
- **Lawson**
- **Progress**

For immediate consideration, send your resume and salary requirements to: **Gordon Food Service, P.O. Box 2066, Grand Rapids, MI 49501; fax: 616-717-9075; email: mperszyk@gfs.com**. Visit us on the Web for detailed job descriptions at: **www.gfs.com**. We value workforce diversity.

GFS®
gordon food service

Growth. Performance. Strength.

CareerLink
For IT Development Professionals

Pro Shop

Strings and Things

Java's syntax is not as simple as it seems, but with a few tricks and an understanding of its boundaries, it can be almost as simple as it seems

BY DANIEL F. SAVARESE

Even though I've programmed more in Java over the past few years than in systems programming languages, I consider myself language agnostic and, for that matter, platform agnostic. There's at least one thing about every programming language and every operating system that I hate and another that I love. When someone praises the superiority of one system over another, I can usually fire back with a counterexample supporting the opposing view. It's not that any system is inherently better than another. Each is designed with different goals in mind and is therefore better suited for different tasks.

I'll be the first to admit that C++ is fraught with complexity, but I am a strong advocate of using C++ (or C in cases where the benefits of object-oriented programming are not required) for performance-oriented systems programming tasks. Java still has too large of a footprint and is not fast enough on average to meet the requirements of many systems-level applications. However, other applications, including those found in distributed systems, are not CPU bound, meaning that the overall execution time of the program is not determined by the processor performance. These programs tend to be I/O bound, and these days that usually means network I/O. For network-bound applications, Java's continuing performance problems are not a factor. That is why many, but not all, middleware applications can be successfully implemented in Java.

Performance isn't the only place where programmers criticize different programming environments. The complexity of a programming language is often a sore point. C++ gets knocked around a lot for its syntactic and semantic complexity. Virtual destructors, templates, and operator overloading are frequently cited sore points. Java programmers are especially fond of pointing out Java's simplicity, especially with respect to operator overloading. Understanding what's going on in C++ programs that use a lot of operator overloading can be very difficult. In Java, you always know what's really happening. Well, not really.

Listing 1 Constant String Concatenation When you concatenate two string constants, the compiler merges them into one string.

```
public class Concat1 {
  public static final void main(String[] args) {
    System.out.println("a" + "b");
  }
}

Method void main(java.lang.String[])
  0 getstatic #7 <Field java.io.PrintStream out>
  3 ldc #1 <String "ab">
  5 invokevirtual #8 <Method void
    println(java.lang.String)>
  8 return
```

Java syntax includes several compromises that make it seem like something efficient is happening when it's really not. Let's take string concatenation as an example. Java includes a String class because programs almost always have to manipulate strings. The most common string manipulation operations are supported as methods of the String class, except for those that actually modify the string. Instances of the String class are immutable. If you want to modify a string value, you have to create a new string. The StringBuffer class is provided to avoid the incessant creation of strings. If you need to modify the value of a string repeatedly, you should use the StringBuffer class.

Using StringBuffer objects, repeatedly calling append() or insert(), can be rather awkward. Ultimately, when you want to do something useful with the StringBuffer, you typically have to go through the overhead of converting it to a string with toString(), because most Java methods that use strings require String parameters rather than StringBuffers. Rather than use StringBuffer explicitly, many Java programmers make heavy use of the + concatenation operator. The concatenation operator is an example of operator overloading in Java. Even though Java does not support programmer-defined operators or overloading, it incorporates overloaded + and += operators. So much for Java being simple.

What does the concatenation operator really do? It seems rather inoffensive and extremely convenient. The fact that it is part of Java syntax can lead you to believe that it is somehow a more optimal way of handling strings. A simple Java program that prints out two concatenated string literals followed by its bytecode representation (generated with javap -c) is shown in Listing 1. Interestingly, the compiler preconcatenates the string literals, turning "a" + "b" into "ab" in the compiled byte code. So far, so good. In Listing 2, we concatenate a string constant with a string variable. The truth behind the concatenation operator is revealed. The compiler has created a StringBuffer instance, initializing it with the first operand and then calling append() using the second operand as an argument. Finally the StringBuffer is converted to a String using toString().

The concatenation operator can also be used to concatenate the string representation of a primitive type value with a string. Listing 3 concatenates a string constant with an integer constant. As with the example in Listing 1, the compiler performs the concatenation and inserts a string literal into the byte code rather than rely on a StringBuffer. The experiment is repeated with an integer variable in Listing 4. The compiler again generates a StringBuffer to perform the concatenation. The lesson here is that even though Java appears to be simpler than C++, there is a close binding between its syntax and its runtime libraries that is not always obvious to the programmer. You may think you have a handle on how many objects your program is creating, and later you find that the compiler has created many temporary objects that affect the performance of your program.

Even though Java shares some of the complexity that plagues C++, it is often criticized for the dichotomy it imposes between primitive and user-defined types and its poor support for generic programming. For example, if you store objects in arrays, writing

Listing 2 String Variable Concatenation When a concatenation includes a string variable, the compiler creates a StringBuffer and uses append() to concatenate the strings.

```
public class Concat2 {
  public static final String LABEL = "string: ";

  public static final void main(String[] args) {
    System.out.println(LABEL + args[0]);
  }
}

Method void main(java.lang.String[])
  0 getstatic #10 <Field java.io.PrintStream out>
  3 new #5 <Class java.lang.StringBuffer>
  6 dup
  7 ldc #1 <String "string: ">
  9 invokespecial #8 <Method java.lang.StringBuffer(
    java.lang.String)>
 12 aload_0
 13 iconst_0
 14 aaload
 15 invokevirtual #9 <Method java.lang.StringBuffer append(
    java.lang.String)>
 18 invokevirtual #12 <Method java.lang.String toString()>
 21 invokevirtual #11 <Method void println(
    java.lang.String)>
 24 return
```

Listing 3 Strings and Integer Constants When you concatenate a string with an integer constant, the compiler merges them into one string.

```
public class Concat3 {
  public static final String LABEL = "string: ";

  public static final void main(String[] args) {
    System.out.println(LABEL + 1);
  }
}

Method void main(java.lang.String[])
  0 getstatic #8 <Field java.io.PrintStream out>
  3 ldc #2 <String "string: 1">
  5 invokevirtual #9 <Method void println(java.lang.String)>
  8 return
```

Listing 4 Strings and Integer Variables When you concatenate a string with an integer variable, the compiler creates a StringBuffer to concatenate the strings.

```
import java.util.*;

public class Concat4 {
  public static final String LABEL = "string: ";

  public static final void main(String[] args) {
    long i = System.currentTimeMillis();
    System.out.println(LABEL + i);
  }
}

Method void main(java.lang.String[])
  0 invokestatic #10 <Method long currentTimeMillis()>
  3 lstore_1
  4 getstatic #11 <Field java.io.PrintStream out>
  7 new #5 <Class java.lang.StringBuffer>
 10 dup
 11 ldc #1 <String "string: ">
 13 invokespecial #8 <Method java.lang.StringBuffer(
    java.lang.String)>
 16 lload_1
 17 invokevirtual #9 <Method java.lang.StringBuffer
    append(long)>
 20 invokevirtual #13 <Method java.lang.String toString()>
 23 invokevirtual #12 <Method void println(
    java.lang.String)>
 26 return
```

Build

Better, Smarter Web Sites

SAVE 42% off the newsstand price!

All the Tools, Tips and Techniques You Need to Create Your Web Site Quickly and Easily!

The WEB Builder Quarterly Design and Development CD is your complete Web development toolkit—everything need to take your Web site to the next level, all in one complete package. For just $29.95 per year, you'll get a new CD every quarter, packed with:

More Hits, Big Savings!

- How-to articles from *Visual Basic Programmer's Journal, WEB Builder, Java™ Pro, Technical Guide to Visual Programming, Enterprise Development, Getting Started with Visual Basic,* and more
- New tools and utilities, complete with the latest versions
- Product reviews, software, and links
- Plus dozens of tips, basics, code... and more!

Subscribe to WEB Builder CD for just $29.95 and watch your hits grow!

It's easy to subscribe to WEB Builder CD:
E-mail us at webcs@magserv.com
Visit us at http://marketplace.devx.com/moretools.asp
Call us at 1-800-848-5523
Or write to us at P.O. Box 15025, N. Hollywood, CA 91615

WEB Builder CD is a product of Fawcette Technical Publications, Inc. All product names herein are the property of their respective owners.

FTP FAWCETTE TECHNICAL PUBLICATIONS

Listing 5 Generic Array Resizing Even though reflection is inefficient, it can be used to create a generic array resizing method that handles arrays of either primitive or Object-derived types.

```
import java.lang.reflect.*;

public class ArrayResize {
  public static final Object resizeArray(Class clss, Object
    src, int newSize) {
    Object newArray = Array.newInstance(clss, newSize);
    System.arraycopy(src, 0, newArray, 0,
      Math.min(Array.getLength(src), newSize));
    return newArray;
  }

  public static final void main(String[] args) {
    int[] intArray = { 10, 9, 8, 7, 6, 5 };
    int[] newArray;
    int i;

    newArray = (int[])resizeArray(
      Integer.TYPE, intArray, 10);

    for(i = 0; i < intArray.length; ++i)
      System.out.print(intArray[i] + " ");
    System.out.println();

    for(i = 0; i < newArray.length; ++i)
      System.out.print(newArray[i] + " ");
    System.out.println();

    newArray = (int[])resizeArray(
      Integer.TYPE, intArray, 3);

    for(i = 0; i < newArray.length; ++i)
      System.out.print(newArray[i] + " ");
    System.out.println();
  }
}
```

a generic algorithm that manipulates both arrays of objects and arrays of primitive types is difficult. Even when you use container classes, you can only work around the problem by using wrapper classes such as Integer and Float.

I was recently asked if there was a generic way to resize an existing array and initialize its undefined values to null or zero. C programmers are used to manipulating raw memory and will either use realloc() or reallocate memory themselves and do a memcpy() to copy the existing array values. In Java, you typically want to use a Vector or ArrayList if you need to use a resizable array. But you can't use them effectively for primitive types. In addition, for certain applications, indexing directly into an array is more efficient. The natural reaction to this situation can be to give up and decide that Java just isn't up to the task and can't handle generic array operations well. This would be giving up too soon.

Java may appear to be simpler than C++, but there is a close binding between its syntax and its runtime libraries that is not always obvious to the programmer

One of the great tricks you can pull off in Java is demonstrated in Listing 5. Every Java type is represented by a Class object whose methods can be inferred through a reflection mechanism. Even though it is inefficient to do this, it allows for the complete dynamic binding of classes and their methods. Arrays themselves are objects. They can be manipulated in the most generic way through the java.lang.reflect.Array class. A generic method called resizeArray() that will resize an array of any type, including primitive types, preserving its original values to the extent that the new size of the array will allow, is defined in Listing 5. Saying that the array is resized is misleading. A new array is created and the old values copied into it. Given a Class object and a size, the Array.newInstance() method will create a new array for a given class. The Array class also allows you to index into an array of arbitrary type and retrieve values, be they primitive or Object-derived.

One of the keys to being an effective programmer is to understand the boundaries of your chosen implementation language. Manipulating strings in Java doesn't have to be expensive if you know when to use character arrays or StringBuffers instead of the String class. Some, but not all, array operations can be performed generically, encompassing both primitive and user-defined types. It can be more difficult to do certain things in Java than in another language, and vice versa. We're at a point now where Java has been around long enough that it's just another tool in our implementation bag. I won't build a pedal-to-the-metal, high-volume transaction processing system in Java, but I'll go ahead and design a log analysis and networked reporting system with it. I just won't rely on the compiler to manage my strings for me. ■

Daniel F. Savarese is vice president of software development at WebOS, Inc., a provider of Internet application infrastructure software and services. He has worked as a senior scientist at the California Institute of Technology and director of Java product development at ORO, Inc. Daniel is the original author of the Jakarta-ORO text processing packages and the NetComponents network protocol library. He is also a coauthor of *How to Build a Beowulf*. He can be reached at dfs@savarese.org.

Go Online

Use the following DevX Locator+ codes at www.java-pro.com to go directly to these related resources.

JP0103 Download all of the code for this issue of *Java Pro*.

JP0103PS Download the code for this article.

JP0103PS_T Read this feature online. DevX Premier Club membership is required.

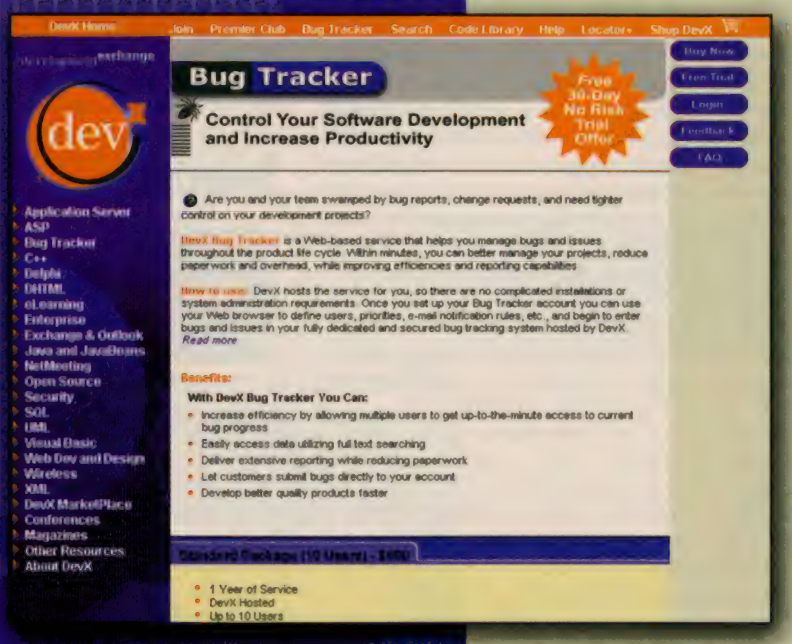
Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

Bug Tracker

Free Trial

Swamped by Bug Reports and Change Requests?

Control Your Development Projects Now with **DevX Bug Tracker**


Bug Tracker
Control Your Software Development and Increase Productivity
Free 30-Day No Risk Trial Offer
Are you and your team swamped by bug reports, change requests, and need tighter control on your development projects?
DevX Bug Tracker is a Web-based service that helps you manage bugs and issues throughout the product life cycle. Within minutes, you can better manage your projects, reduce paperwork and overhead, while improving efficiencies and reporting capabilities
How to use: DevX hosts the service for you, so there are no complicated installations or system administration requirements. Once you set up your Bug Tracker account you can use your Web browser to define users, priorities, e-mail notification rules, etc., and begin to enter bugs and issues in your fully dedicated and secured bug tracking system hosted by DevX.
Read more
Benefits:
With DevX Bug Tracker You Can:
Increase efficiency by allowing multiple users to get up-to-the-minute access to current bug progress
Easily access data utilizing full text searching
Deliver extensive reporting while reducing paperwork
Let customers submit bugs directly to your account
Develop better quality products faster
1 Year of Service
DevX Hosted
Up to 10 Users

This new Web-based service helps you:

- manage bugs and issues throughout the product life cycle
- reduce paperwork and overhead
- improve efficiencies and reporting

And, because DevX hosts Bug Tracker for you there are no complicated installations or system administration requirements.

Sign Up Today for your FREE 30-DAY TRIAL

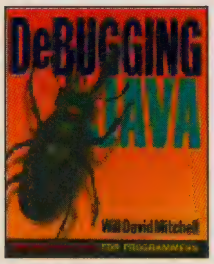
Go to www.devx.com/freetrial

or call 1-888-963-3389
1-650-566-2000 (outside the USA)

DevX, 913 Emerson Street, Palo Alto, CA 94301

DevX, and DevX Bug Tracker
are trademarks of DevX.com, Inc.

Book Reviews



Debugging Java: Troubleshooting for Programmers
by Will David Mitchell
Berkeley, CA: Osborne/McGraw-Hill
ISBN 0072125624
500 pages
Price: $39.99

Some books deserve special attention and belong on every serious programmer's shelf. If you develop programs in Java, Will David Mitchell's *Debugging Java: Troubleshooting for Programmers* is one of those books you need. Most of the Java books in print today center on a particular set of APIs or the infrastructure you need to develop Java solutions. This book centers on techniques that make you more productive. Despite Java's inherently safer programming constructs, poor software development can still lead to serious errors in your code. By exploring what can go wrong with a program from multiple angles, this book can make you a more efficient, safer, more relaxed, and better-informed developer.

No debugging book would be complete without a thorough exploration of where bugs come from and how you can avoid them. Debugging Java is divided into three parts and five appendices. The first part centers on writing bug-free code and works toward making good use of preventative techniques and the proper application of exception handling. Chapter 4 introduces something the author calls "risk factor analysis," which may be as practical as it is entertaining. It offers multiples you can apply to coding estimates, based on the amount of knowledge you have when you provide the estimate. Part II centers on removing existing bugs. By applying the lessons from Part I you can reduce this load, but it's Part II that helps you find and kill the bugs that remain. Mitchell does a great job of classifying the types of bugs you are likely to find and helping shed light on the tools that help you deal with each type. Debuggers play a big role in finding bugs and Chapter 8 can lead you to a few of the good ones. This part of the book also covers testing, disciplines that can help you, and multithreaded bugs, among other things.

The book wraps up with Part III, which is really a single chapter on performance. Compared to other books out there, this is not the book's greatest strength. However, I found the chapter both authoritative and useful.

If you take Java programming seriously and you want to improve your skills, you can't take debugging lightly. What's more, prevention is clearly the way to make your projects more successful. Understanding the variety of bugs that can plague your code can make them easier to avoid and conquer if they slip through your fingers. *Debugging Java: Troubleshooting for Programmers* is one of those books that should be required reading in any Java shop. You'll benefit by reading it, even if you only work with Java occasionally.

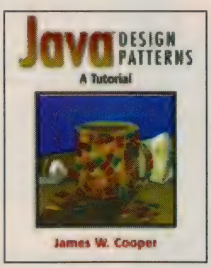
Enterprise Java Performance
by Steven L. Halter et al.
Upper Saddle River, NJ:
Prentice Hall PTR
ISBN 0130172960
432 pages
Price: $49.99

Server-side Java has quickly become the primary choice of developers when it comes to commercializing products and services or upgrading legacy systems. Large systems need close attention, and innovative solutions are fraught with potential pitfalls that must be overcome. *Enterprise Java Performance* is likely to be an asset to any Java programmer, especially one working on high-performance solutions. This book takes a good look at the process of optimizing and avoiding bottlenecks, from design to testing. It covers various tools and technologies that can make you a better programmer and helps clarify some of the lesser known issues in enterprise Java solutions.

The book is divided into six parts. Part I takes a high-level view of performance as a whole, talking about different types of performance issues and classifying problems into physical limitations, application models, and low hanging fruit. This section also takes a quick look at profiling tools and performance measurements, as well as common problem areas. Part II looks at the physical limitations programmers have to deal with, including issues surrounding remote processing and bottlenecks. I found this information invaluable and the treatment of bottlenecks is especially relevant to multithreaded applications, regardless of whether they are client or server solutions.

Part III takes a great look at benchmarking, which to my knowledge is never explored that well in any existing Java book. Both low-level JVM performance and application-level benchmarking are explored. Part IV takes a look at application environments. To provide relevant studies, Steven L. Halter and Steven J. Munroe look at IBM's SanFrancisco framework, EJB, CORBA, and even Jini performance, which is becoming increasingly important in enterprise applications. Part V looks at application models, exploring issues like database connectivity, middleware interaction, memory leaks, and garbage collection, to name just a few. Part VI completes the treatise with a focus on scaling, system tuning and large-scale effects, such as memory, processor, and i/o constraints and analysis.

I found *Enterprise Java Performance* incredibly valuable in my own environment and can't help but recommend it highly. As a Java developer, you need to know which classes cause the most problems; how to deal with multithreading and synchronization issues; how to manage memory, CPU, and I/O resources; how to measure performance; and how to interpret benchmarks, understand the implications of cross-machine interactions, and see how all these issues affect architectures like EJB, CORBA, and Jini. In all, this is a book worth its weight in gold to the right reader, and clearly worth more than the price to you even if you only learn a few of the fundamental keys to better performance in your own development efforts. I recommended it highly.



Java Design Patterns: A Tutorial
by James W. Cooper
Reading, MA: Addison-Wesley
ISBN 0201485397
352 pages
Price: $44.95

Wiley Computer Publishing made a valiant attempt at tackling the Pattern idioms directly with Java in Mark Grand's *Patterns in Java, Volume 2*, but James W. Cooper's *Java Design Patterns: A Tutorial* from Addison-Wesley delivers. Unlike its competition, this book clarifies difficult issues and provides well-described examples that lead the user to a clear understanding of the concepts involved.

I can't overstate this. It's difficult to convey these ideas in short order and leave the reader with few questions at the end of each description. After reading this book, you should be in a position to explain a pattern to others without feeling uncertain about the possible interpretations to which it might have been subjected. For any serious Java programmer, this book is a great investment.

The book is divided into 36 chapters and six major sections. The first

section is dedicated to introducing patterns, with a little historical context, and UML, which forms the basis for illustrating each of the patterns covered later. The main patterns are documented in the next three sections, covering Creational, Structural, and Behavioral patterns. The book closes with a section that looks at the JFC (Java Foundation Classes), highlighting the use of various patterns in the framework. The last section takes a quick look at a few case studies to help clarify a few concepts a little further.

Each of the chapters that focus on a particular pattern—effectively the bulk of the book—starts by describing the pattern in unambiguous terms, drawing a UML representation and tying the pattern to one or more specific examples. In each case, code is developed along the way to produce a practical implementation that clearly demonstrates the use of a given pattern. I found each case readily accessible and unpretentiously explained in a way that makes the material both engaging and clearly explored. The code is intentionally kept uncomplicated but has enough substance to make the topic matter easy to understand as it's applied in context.

Java Design Patterns can make you much more effective both as a programmer and an architect. Using the same terminology to communicate more effectively leads to better software and a better sense of tried-and-true practices that lead to stronger implementations. Since it directly applies the quintessential patterns judged central to development, you can't really go wrong picking up this book.

Essentials of the Java Programming Language: A Hands-On Guide
by Monica Pawlan
Reading, MA: Addison-Wesley
ISBN 0201707209
320 pages
Price: $34.95

When I am asked to recommend a good book for beginners, I typically have to inquire about their existing programming skills and knowledge before making a suggestion. One of the questions I ask is whether the reader can handle a 500-page (or more) book. Although there may not be many Java books in print that are really appropriate for light readers, *Essentials of the Java Programming Language: A Hands-On Guide* may well be the answer for numerous readers interested in a light, yet meaningful, visit to the world of Java programming.

Monica Powlan's book, divided into 14 lessons, provides a high-level overview of particular areas of utility in the Java language. It teaches you how to install the JDK, compile programs, and deal with some basics of program construction before showing you how to tackle small application components that can be assembled later in the book.

Powlan uses a hands-on approach to help you build a simple applet, including covering user interfaces with Swing and building a servlet. She discusses file access permissions and database access, and ties communication together using both RMI and socket connections. High-level topics and improvements to the sample application are revisited in later chapters to polish things up, add internationalization, and package it into JAR distributables. All of this is made accessible to anyone who's had some experience with programming in any language—that is, someone who knows the fundamental concepts but doesn't know much about Java. The multitiered application developed in these pages represents a microcosm of larger Web apps and enables you to learn the key issues surrounding this kind of development. This is a good approach for a lot of reasons—not the least of which is the overall understanding of multi-tiered application development, including internationalization and deployment.

Though you won't get any tips on scalability, and the application is oversimplified, you will be well served by

Product Showcase

JAVAPro

Bar Code JavaBeans & Fonts

JavaBeans, Applets, Barcode Fonts and HTML embedded fonts for Code 39, POSTNET, Code 128, UPC, PDF417, OCR, MICR and many others for integration into your application. We also offer barcode UFLs for Crystal Reports, MS Office Macros and Java Source Code.

http://www.IDAutomation.com

This is your Showcase Section for java tools and applications. If you are interested in promoting your tool or product to Java developers, please call 650.833.7100.

FTP FAWCETTE TECHNICAL PUBLICATIONS

Java is a registered trademark of Sun Microsystems, Inc.

JAVAPro

Subscribe Now!

Java Pro is the progessional developer's independent source for information about building serious Java applications.

Java Pro takes you beyond the hype by delivering hands-on, code-intensive, product-oriented information that helps you harness Java's unique capabilities in your enterprise-wide development efforts. Each month, Java Pro tackles tough development issues and provides unbiased, no-nonsense coverage of the latest tools on the market.

Call today to subscribe to Java Pro. You'll pay the low rate of $29.95 for a full year-thirteen issues in all. That's a savings of more than 61% off the newsstand rate. To subscribe now call 800-848-5523. Outside the U.S. call 650-833-7100. Or, order online at www.java-pro.com.

FTP FAWCETTE TECHNICAL PUBLICATIONS

Fawcette Technical Publications
209 Hamilton Ave.
Palo Alto, CA 94301

Java is a registered trademark of Sun Microsystems, Inc.

Book Reviews

this book. Nothing is assumed about previous Java knowledge, and a minor understanding of programming is probably sufficient to get what you need out of this book. A lot of information is packed into fewer than 300 pages, but it is very accessible, and the resulting application is both non-trivial and comprehensive enough to learn much about real-world Java development.

Java Programmer's Reference

by Grant Palmer
Chicago, IL: Wrox Press Inc.
ISBN 1861004222
1,200 pages
Price: $34.99

I have to admit to fairly ambiguous feelings about Grant Palmer's book, and cannot decide whether it should be recommended or trashed. In the end, of course, whether you like it or not depends on what purpose you have for a book. This is a book that does many things well, but also tends to fall short in living up to its apparent title. *Java Programmer's Reference* does provide a useful reference for many of the Java APIs, but not all of the standard packages are covered. Those packages that are covered are covered fairly well; however, the book could have included more examples of how to use the more important classes.

This book provides nice explanations of the fundamentals, presented in an easy-to-read, accessible format that allows you to learn what you need and clear up any questions you might have. This is precisely what you need from a reference book. The major packages are explored, each in their own chapter, with good coverage of the java.lang, java.util, java.io, java.net, java.awt, java.awt.event, java.applet, and java.beans packages. Many of the Swing packages (prefixed by javax.swing) are covered, including the main Swing package, border, table and tree. What's explained seems well addressed, if perhaps a little terse by virtue of the book's reference format.

What's not covered in these pages may well be the book's greatest weakness. Most of the java.awt subpackages like color, datatransfer, dnd (drag and drop), font, geom (geometry) and im (input) methods, image, and print are simply not there. Nothing about RMI is mentioned, other than to let the reader know it won't be covered. While the java.beans package is documented, none of the reflect (reflections), ref (reference), or beancontext packages are covered. The java.math, all of the security APIs, and many of the Swing support packages—including the javax.swing.text package—are missing. It might be understandable that CORBA support is not covered, or even the recently added Java Naming and Directory Interface (JNDI), but there is much that should have been included.

If most of your work keeps you right in the center of Java development with only occasional forays into graphic user interfaces or standard extensions, you may like this book. The format is very accessible and the information is well organized, although somewhat less than comprehensive. You won't be able to tell in which version a given method was introduced, but then the lack of clutter may be a selling point, depending on how you look at it. I loved the way each covered package was divided into logical subsections with groupings of related classes. This alone made the information more navigable. Overall, this is an expensive book for what it delivers, and there are many other comprehensive references in the $20 range that will more aggressively compete for the same readers. You may be one of them. ■

Claude Duguay is vice president of engineering at 360.com. Having developed software for more than 20 years, he is passionate about Java and reads about it incessantly, writing about it as often as possible. You can reach him at claude@360.com.

Advertiser Index

Allaire Corporation 888-939-2545 www.allaire.com/download	19	**Fiorano** www.fiorano.com	37	**Sitraka Software** 888-361-3264 www.sitraka.com/j2ee-javapro	1	**Showcase**	
Amazon.com www.amazon.com	51	**InformIT** www.informit.com	26	**Sitraka Software** 888-361-3264 www.sitraka.com/javapro	C3	**ID Automation.com** www.IDautomation.com	77
Appstream 866-AppStream www.appstream.com	27	**Infragistics, Inc.** 800-231-8588 www.infragistics.com	2, 3	**Sitraka Software** 888-361-3264 www.sitraka.com/javapro	C4	**CareerLink**	
Aritistic Systems www.artisticsystems.com	23	**Inprise** 800-331-0877 www.borland.com	48	**ThunderLizard.com** 800-221-3806 thunderlizard.com/data	61	**Gordon Food Service** www.gfs.com	70
Compuware Corporation 800-4-NUMEGA www.compuware.com/numega	55	**InstallShield Software Corp.** www.onesetup.com	33	**Tidestone Technologies** 800-884-8665 www.tidestone.com	43	**CareerLink Network**	68-69
Developmentor 800-699-1932 www.develop.com	46	**Java Pro Magazine Subscriptions** 800-848-5523 www.java-pro.com	25	**Togethersoft Corporation** 919-833-5550 www.togethersoft.com/jp/	C2		
DevX.com 888-963-3389 www.devx.com	62	**LOOX Software** 800-684-5669 www.loox.com	47	**Unify Corporation** 800-GO-UNIFY shop.unify.com	39		
DevX Bug Tracker 888-963-3389 www.devx.com/bugtracker/	75	**Metrowerks, Inc.** 800-377-5416 www.metrowerks.com	52	**Visual C++ Developer's Journal Magazine Subscriptions** 800-848-5523 www.vcdj.com	57		
DevX eLearning 888-963-3389 www.devx.com/elearning	67	**Parasoft Corporation** 888-305-0041 www.parasoft.com/jpro3	44	**Web Builder CD** 800-848-5523 marketplace.devx.com	73		
DevX MarketPlace 888-963-3389 marketplace.devx.com	65	**Poet Software** 800-950-8845 www.poet.com	12	**WebGain** 888-822-3409 www.webgain.com	59		
Embarcadero Technologies 800-811-2784 www.gdpro.com	15	**Progress Software** 800-989-3773 x325 www.sonicmq.com/jp301	17	**XML Magazine Subscriptions** 800-848-5523 www.xmlsubs.com	79		
Empirix www.bean-test.com/jvp	31	**RadView** 888-RadView www.radview.com	10	**Zero G** 888-512-7771 www.ZeroG.com	4		
Exchange & Outlook Multimedia Resource 800-848-5523 exchange.devx.com	38	**Rational Software** www.rational.com/javapro	8				
		SilverStream 888-823-9700 www.silverstream.com/onlineseminars	6				

Product Information

- To get immediate details on products advertised in this issue, you can simply log onto our online services at **http://infoasap.devx.com**.
- This provides a list of direct links to this month's advertiser's Web sites.

Advertising Sales Contacts

- **Shana Deane, Advertising Director** 413.665.6910, sdeane@fawcette.com
- **Beatrice Stonebanks, Southern Regional Sales Manager** 650.833.7163, bstonebanks@fawcette.com
- **David Lyman, Eastern Regional Sales Manager** 978.465.2351, dlyman@fawcette.com
- **Janice Mahler, Northwestern Regional Sales Manager** 949.387.6684, jmahler@fawcette.com
- **Cecile Pendleton, Account Manager** 650.833.7120, cpendleton@fawcette.com

This listing is provided as a courtesy to our readers and advertisers. The publisher assumes no responsibility for errors or omissions.
Sales Department: 209 Hamilton Avenue • Palo Alto, CA, USA 94301-2500 • 650-833-7100

XML Solutions and Strategies:

Get them here first!

How to Choose an XML Ser
XML
MAGAZINE
.NET=XML
How XML Will Move Microsoft Beyond Windows
Apps to Web Services with SOAP
Serving XML with JavaServer Pages
Expert Voices: .NET's Wascha on XLANG, Hejlsberg on C#
Cocoon Separates Logic from Layout
Microsoft .net

FREE Subscription!

XML Magazine is the only independent publication devoted 100-percent to the latest information about using XML for enterprise-level challenges. Created by the same experts who gave you *VBPJ, Exchange & Outlook* and *Java Pro,* this is the one indespensible XML resource that you can't afford to be without. Published every other month, you'll get practical, hands-on articles, tips and enterprise-level solutions, including:

- Pragmatic techniques for developing with XML
- Cutting-edge analysis of industry trends
- Independent product reviews and new product announcements
- Columns by the industry experts who design the technology
- Complete source code examples

Sign up today for your FREE subscription. Complete the postage-paid card in this issue.

FTP FAWCETTE TECHNICAL PUBLICATIONS

209 Hamilton Avenue, Palo Alto, CA 94301-2500 • Phone: 650.833.7100 • Fax: 650.853.0230

©2000, FTP, Inc. XML is a trademark of MIT. Publisher reserves the right to determine qualifications for free subscriptions.

The Dot Gets Slashed

by Sean Gallagher
Editorial Director
sgallagher@fawcette.com

I'd like to express my deepest condolences to those of you who may recently have experienced the "reverse-wealth effect." If you've been taking a significant piece of your compensation as stock options, the sudden decompression of the dotcom bubble may have come as something of a shock to you. The rest of us promise to try not to tell you, "I told you so."

Last year, *Java Pro*'s salary survey ("How Does Your Career Compare?" June 2000) found that Java developers had experienced an 11% annual increase in total compensation. As we prepare to start this year's survey, we are looking at a totally different world than the one we saw last spring. Fortunately, the bursting of the dotcom bubble doesn't seem to be affecting the Java job market much.

In the short term, at least, it may even make lives a little easier for some, as long-unfilled Java slots get filled by freshly released talent. Free agents may lose a few clients, but they'll pick up some others. All will remain sweetness and light—for a while. But things will change soon. Even if the economy can continue to go gangbusters this year, the nature of the "new economy" is undergoing a fundamental change. The bloom is off the rose.

This isn't just the concern of day-traders and advertising execs—though I'm sure Sun is reconsidering its "Dot in Dotcom" campaign right now. The backlash against dotcom startups caused by the failures of seemingly hundreds of "pure play" companies with marginal business plans will be felt by the rest of us as well.

At some companies, it will affect the kinds of projects we're offered—they may not be as cool, or as aggressive in exploiting new technologies. It will affect the level of commitment to those projects by management and the ability to wring resources free to work on them, too.

This doesn't mean money is going to dry up. It just means that it'll be pooling in different places. Java is in a position to be the dominant technology in the race to further integrate businesses with Web services and other Internet-based technologies. There will be more Java jobs in wireless and embedded development and maybe even in the desktop application space.

But the really exciting stuff won't be happening at creatively spelled dotcoms in VC-funded digs in Manhattan or San Francisco. It's going to be happening (horror of horrors) in the land of bricks and mortar, at companies with bleeding-edge names like General Electric. It won't necessarily be cool, it definitely won't be glamorous, and it won't be creating another "Generation Equity." But it will bring the whole old economy into the new economy and permanently change the way trillions of dollars' worth of business is done.

Java won't be the only player in this new landscape by any stretch of the imagination. Microsoft is trying to pull its .Net platform together, and corporate customers are already using pieces of it. But as the W3C works to build a standard for XML-based remote procedure calls from the groundwork laid by XML RPC and SOAP, Microsoft may find itself the victim of another spin on "embrace and extend"—as Java developers find ways to exploit what works in the Windows world through remote interfaces without having to develop on the platform.

Java can be the dominant technology in the race to integrate businesses with Web services and other Internet-based technologies

Perhaps the longest lasting impact of the dotcom boom will be the way it has changed the culture of corporate IT. The early success of flash-in-the-pan companies has made many traditionally staid corporate behemoths re-examine their own ways of doing business, and the value of corporate software professionals. Competing with dotcoms for talent has meant that some of the more progressive incentives for employment have found their way into the corporate mainstream.

So, rejoice, dotcommers! Your forfeited stock options have not died in vain. You may even find life in the corporate world tolerable. Unfortunately, I can't guarantee you won't have to wear a tie. ■

Go Online

Use the following DevX Locator+ code at www.java-pro.com to go directly to this related resource.

JP0103SG_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www. DevX.com to subscribe to the Premier Club.

© Copyright 2000 by Sitraka Inc. All rights reserved. Sitraka, the Sitraka logo and JClass are trademarks of Sitraka Inc. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in US and other countries. All other products are trademarks or registered trademarks of their respective companies.

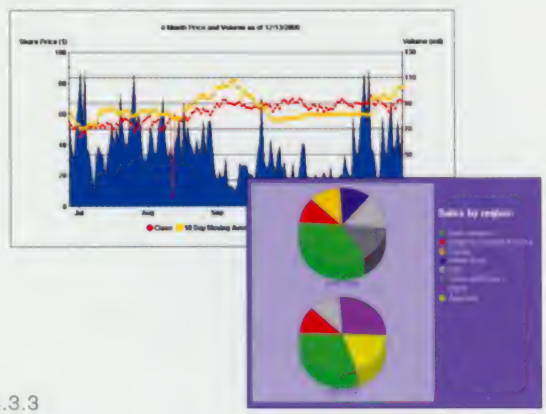
From the leader in Java GUI components...

JClass ServerChart, pure innovation in server-side development.

JClass™ ServerChart

Introducing the first truly powerful solution for thin-client data visualization: JClass ServerChart. With a wide range of chart and graph types and built-in image map tag generation, JClass ServerChart gives end users of your application real-time access to business-critical data in an intuitive and highly interactive format.

Designed from the ground up for server-side deployment, JClass ServerChart integrates with major Java-compliant Web and application servers, and leverages servlet and JSP technologies to ensure maximum enterprise scalability.



- multiple business and scientific chart types
- fully customizable
- populate charts with data from files, sockets, URLs, XML-based data, or customized sources
- works in headless environments when used with eTeks' PJA toolkit
- integrates with major Web and application servers and IDEs
- support for Win ME/98/NT/2000; Solaris 2.8, Redhat Linux 6.2, AIX 4.3.3
- support for JDK 1.1.8, 1.2.2, 1.31

Download a free eval now

www.sitraka.com/javapro

Formerly KL Group 1 888-361-3264 | Europe +31(0)20 510 67 00

sitraka software

the Java™ advantage

986 brokers want that trading application update and they want it now

DeployDirector™

Java application deployment and management solution

If you're in charge of deploying and updating Java applications, you're probably no stranger to pressure. Your deployment deadlines are looming. You have multiple clients, multiple platforms, minimal time and minimal resources. No doubt IT and development are already going full-speed. The network can't handle huge downloads without crippling productivity. Meanwhile, your end users just want to keep working and don't want installation headaches.

Finally, there's a solution out there that makes Java deployment easier. Much easier. Sitraka Software's DeployDirector delivers on the promise of Java by providing enterprise environments with a seamless solution for deploying and updating Java applications and applets. It's fast, scalable, and cost-effective. And that means you can satisfy at least 986 people right now. In fact, make that 987.

Download a free white paper and a trial version of DeployDirector at:

www.sitraka.com/javapro

Formerly KL Group 1 888-361-3264 | Europe +31(0)20 510 67 00

sitraka software

the Java™ advantage

© Copyright 2000 by Sitraka Inc. All rights reserved. Sitraka, the Sitraka logo and DeployDirector are trademarks of Sitraka Inc. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in US and other countries. All other products are trademarks or registered trademarks of their respective companies.

JAVA Pro

YES ☑ Send me one year (13 issues) of *Java™ Pro* for only $29.95

Name _____

Address _____

City _____ State _____

Zip _____ Country _____

E-mail address _____

❑ Bill me ❑ Payment enclosed Charge my: ❑ Visa ❑ Master Card ❑ AMEX

Card # _____

Expiration date _____

Signature _____

One year is 12 issues plus one special issue. Savings based on annual cover price of $77.35. Please allow 4-6 weeks for delivery of your first issue. International subscriptions must be payable in US dollars plus postage: Canada/Mexico add $18/year for surface delivery; all other countries add $44/year for airmail delivery. Canadian GST included.

4192

SAVE 61% SAVE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 Palo Alto, CA

POSTAGE WILL BE PAID BY ADDRESSEE

P.O. Box 54584
Boulder, CO 80323-4584

PASS IT ON.

☐ I want to learn how to control the synchronization between code and models.

☐ Please send me the FREE Java white paper and CD.

Request yours at www.rational.com/javapro or complete and mail this postage-paid reply card.

NAME _____ TITLE _____

COMPANY _____

ADDRESS _____

CITY/STATE/ZIP _____

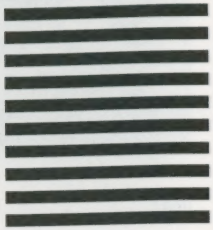
PHONE _____ FAX _____

EMAIL _____

AR-JAVA2-Do

Rational®
the e-development company™

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 9556 LEXINGTON MA

POSTAGE WILL BE PAID BY ADDRESSEE

RATIONAL SOFTWARE
20 MAGUIRE ROAD
LEXINGTON MA 02421-9566